NASA/CP—1999–209235

1999 Flight Mechanics Symposium

Edited by
John P. Lynch
NASA Goddard Space Flight Center, Greenbelt, Maryland

Proceedings of a conference sponsored and held at
NASA Goddard Space Flight Center
Greenbelt, Maryland
May 18–20, 1999

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

May 1999

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076–1320
Price Code: A17

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Price Code: A10

FOREWORD

The papers included here have been presented in summary form at the Flight Mechanics Symposium held May 18-20, 1999, at the NASA Goddard Space Flight Center. For completeness, abstracts have been included for those papers that were presented but unavailable at the time of printing. The papers in this document are presented as received from the authors, with little or no editing.

Table of Contents

Session 1: Attitude Determination, Sensor Calibration, and Computational Techniques

Session 2: Trajectory Design, Autonomous Navigation and Computational Techniques

Session 3: Attitude Control System Design & Simulation I

Session 4: Constellation Design and Formation Flying

Session 5: Attitude Control System Design & Simulation II

Session 6: In-Flight Experience

EXTENDED QUEST ATTITUDE DETERMINATION FILTERING

Mark L. Psiaki
Cornell University

ABSTRACT

The quaternion estimation (QUEST) batch attitude determination algorithm has been extended to work in a general Kalman-filter framework. This has been done in order to allow the inclusion of a complicated dynamics model and to allow the estimation of additional quantities beyond the attitude quaternion. The QUEST algorithm, which works with vector attitude observations, serves as a starting point because it is able to work with a poor (or no) first guess of the attitude. It is able to do this because its nonlinear estimation problem can be solved exactly by solving an eigenvalue/eigenvector problem. This paper's extended version of QUEST uses square-root information filtering techniques and linearization of the dynamics to handle all of the non-QUEST parts of the estimation problem. The remaining QUEST-type part of the problem can be solved by a technique that is an extension of the original QUEST algorithm's eigenvalue/eigenvector solution. The paper shows that two previously-proposed iterative QUEST techniques are special cases of the present algorithm. It also demonstrates the new algorithm's performance on an attitude determination problem that uses star-tracker and rate-gyro measurements to estimate the attitude time history and the rate-gyro biases. The new algorithm is able to converge from initial attitude errors of *180°* and initial rate-gyro bias errors as large as *2,400°*/hour.

INTRODUCTION

Most space missions require knowledge of the spacecraft's attitude. This knowledge is normally derived from on-board sensor data. Possible measured quantities include the Sun direction vector, the Earth's magnetic field vector, the Earth nadir vector, the Earth-limb crossing time of a horizon scanner bore sight, the direction vectors to bright stars, and the differential carrier phase of Global Positioning System signals [1,2]. Some of these quantities contain 2-axis attitude information: a unit direction vector measured in spacecraft coordinates and known in inertial coordinates. Other quantities contain only 1 axis worth of information: the cosine of an angle between a spacecraft referenced vector and an inertially-referenced vector.

In order to derive a spacecraft's full 3-axis attitude, 2 or more attitude measurements must be processed together. A variety of methods exist for processing attitude data. These include geometric-based methods [1], extended Kalman filters [3-8], and a special algorithm known as the quaternion estimation (QUEST) algorithm [9,10]. Geometric-based methods can operate on any type of attitude data, but they do not easily make use of redundant data or complex dynamic models. Kalman filters are excellent at handling multiple redundant sensor signals and at incorporating dynamic models and data that has been measured at different times. Unfortunately, an extended Kalman filter can exhibit sensitivity to the initial attitude guess because it relies on linearizations of the spacecraft's nonlinear measurement and dynamics models. In some situations this sensitivity can cause an extended Kalman filter to diverge [6].

The basic QUEST algorithm solves Wahba's problem [11]. Given a set of known unit direction vectors in inertial coordinates, r_i for $i = 1, \ldots, m$, their measured values in spacecraft coordinates, b_i for $i = 1, \ldots, m$, and their per-axis direction uncertainties (in radians), σ_i for $i = 1, \ldots, m$, the problem is to

$$\text{find:} \qquad A(q) \tag{1a}$$

$$\text{to minimize:} \quad J_{QUEST}\{A(q)\} = \frac{1}{2}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\{b_i - A(q)r_i\}^T\{b_i - A(q)r_i\} \tag{1b}$$

$$\text{subject to:} \qquad q^T q = 1 \tag{1c}$$

where q is the attitude quaternion for the transformation from inertial coordinates to spacecraft coordinates and $A(q)$ is the direction cosines matrix for that same transformation. The formula for $A(q)$ can be found on p. 414 of Ref. 1.

The QUEST algorithm has advantages in comparison to standard extended Kalman filters. One great advantage is that it can be solved exactly by solving an eigenvalue problem [9]. It can never diverge because this solution procedure does not depend on having a first guess. Another advantage is that it explicitly and optimally preserves the attitude quaternion's normalization.

The QUEST algorithm also has disadvantages in comparison to extended Kalman filters. One disadvantage is that it can deal only with vector-type measurements, not with cosine-type measurements. This limits its use to missions where the attitude measurements are all vector-type measurements. A more significant disadvantage of the QUEST algorithm is that it can deal only with very simple dynamic models. The attitude rate time history must be input to QUEST in order for it to use data that has been measured at different times [10]. If this attitude rate time history is derived from rate-gyro measurements, then they cannot have significant biases, which constitutes a severe limitation.

A related disadvantage is the QUEST algorithm's inability to estimate anything other than the attitude quaternion. In its original form, it cannot be used to estimate quantities such as sensor misalignments, rate-gyro biases, or other typical filter states. Therefore, QUEST cannot be used as part of a general attitude determination filter. A linearized version of the QUEST measurement equation can be used in a generalized filter [5], but linearization makes any such algorithm prone to diverge if the initial attitude uncertainty is large.

It would be a great advantage if the QUEST algorithm could be extended to handle an arbitrary dynamic model and the estimation of states other than the attitude quaternion. With such extensions, the QUEST algorithm could be applied using Euler's equations to estimate the attitude rates or using realistic rate gyros to measure them. If Euler's equations were used, then the attitude rates would be estimated as part of the filter state. If rate gyros were used, then the filter's state could include estimates of the rate gyros' biases.

References 12 and 13 document an attempt to extend QUEST to include estimation of other parameters. That approach is based on solving the following extended problem:

find: q and x (2a)

$$\text{to minimize:}\quad J(q,x) = \frac{1}{2}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\{b_i(x) - A(q)r_i(x)\}^T\{b_i(x) - A(q)r_i(x)\} + \frac{1}{2}(x - x^0)^T W^0 (x - x^0) \tag{2b}$$

$$\text{subject to:}\quad q^T q = 1 \tag{2c}$$

where x is a vector of additional parameters, x^0 is it's *a priori* value, and W^0 is a symmetric positive-semidefinite weighting matrix. A problem with dynamics and rate-gyro measurements can be posed in this form [13]. The rate-gyro biases are estimated as part of the x vector.

Reference 12 develops a solution algorithm for problem (2a)-(2c). It works with guesses of the optimal x and solves exactly for the corresponding optimal q by using the QUEST procedure. An outer loop improves the x guesses by numerical iteration. The algorithm is a batch algorithm, and it showed no significant advantages in comparison to a standard batch algorithm when compared on a test problem [13].

This paper has two goals. One is to extend the QUEST algorithm to include arbitrary dynamics and additional estimated states while retaining QUEST's measurement-error cost function and its explicit constraint of the quaternion norm. The other goal is to develop an iterative QUEST filter rather than a batch filter, one that functions as much as possible like an extended Kalman filter while retaining the good features of the QUEST algorithm.

Achievement of these goals will constitute an advance over the work of Ref. 12. That paper's algorithm cannot handle arbitrary dynamics, and it requires numerical iteration to converge to an estimate of the auxiliary x vector. The new algorithm uses extended-Kalman-filter-type stage-to-stage iterations to achieve convergence to its x estimate. This type of iteration is normally much faster than batch-filter iteration because this type computes problem function gradients only once per stage.

This paper also represents an advance in the area of preserving quaternion normalization within the context of an extended Kalman filter. References 4, 5, and 7 develop special techniques to preserve quaternion normalization within a linearizing extended Kalman filter. They do not explicitly enforce the quaternion normalization constraint during the attitude update. Rather, they use ways that implicitly enforce the constraint [5], or they develop ways to re-normalize the quaternion after the update [4,7]. The present paper explicitly enforces the quaternion normalization in an optimal manner.

The extended QUEST algorithm is presented and analyzed in the four main sections of this paper. The second section reviews the QUEST algorithm and its associated quadratically-constrained quadratic optimization procedure. It then presents and solves an extended quadratically-constrained quadratic problem that is compatible with the extended QUEST algorithm. The third section presents the extended QUEST filtering problem statement and the algorithm that solves the problem. The fourth section shows how to compute the estimation error covariance of the filter. The last main section presents test-case results that are based on data from a simulated truth model.

REVIEW AND EXTENSION OF QUEST SOLUTION MATHEMATICS

Original QUEST Solution

QUEST's efficient solution of Wahba's problem hinges on the fact that the cost function in eq. (1b) can be written as a quadratic form in q:

$$J_{QUEST}\{A(q)\} = \sum_{i=1}^{m} \frac{1}{\sigma_i^2} + \frac{1}{2} q^T H_{meas} q \tag{3}$$

where the symmetric Hessian matrix in eq. (3) is [9,10,12]:

$$H_{meas} = \sum_{i=1}^{m} \frac{1}{\sigma_i^2} \begin{bmatrix} \{I(b_i^T r_i) - r_i b_i^T - b_i r_i^T\} & -(b_i \times r_i) \\ -(b_i \times r_i)^T & -b_i^T r_i \end{bmatrix} \tag{4}$$

The minimization of $J\{A(q)\}$ in eq. (3) subject to the quaternion normalization constraint, $q^T q = 1$, constitutes a quadratically-constrained quadratic program. If one adjoins the quaternion normalization constraint to the cost function using the Lagrange multiplier $\lambda/2$, then differentiation of the resulting Lagrangian function with respect to q leads to the following optimality necessary condition:

$$(H_{meas} + \lambda I)q = 0 \qquad \text{or} \quad H_{meas}\, q = -\lambda\, q \tag{5}$$

From the right-hand version of eq. (5), it is plain that q is a normalized eigenvector of H_{meas} and that $-\lambda$ is the corresponding eigenvalue. The optimal solution to Wahba's problem is achieved when q corresponds to the $-\lambda$ value that is the smallest (the most negative) eigenvalue of H_{meas}.

A Generalized Quadratic Program

The extended QUEST filter needs to be able to solve a slightly more general quadratically-constrained quadratic program:

find: $\quad q$ (6a)

to minimize: $\quad J(q) = \frac{1}{2} q^T H q + g^T q \; + constant$ (6b)

subject to: $\quad q^T q = 1$ (6c)

In this formulation, H is the cost function's Hessian matrix, and g is the cost function's gradient vector at $q = 0$.

There are two differences between this problem and the quaternion optimization form of Wahba's problem. Both of them arise from the inclusion of *a priori* information at the given sample instant. The first difference is that there is a linear cost term, $g^T q$. The second difference is that the Hessian matrix, H, is no longer the H_{meas} matrix given in eq. (4). Instead, H will be a combination of H_{meas} and an *a priori* term.

Solution of the Generalized Quadratic Program

Problem (6a)-(6c) can be solved by forming a Lagrangian and deriving optimality necessary conditions [14]. In this case, eq. (5) generalizes to become:

$$(H + \lambda I)q + g = 0 \tag{7}$$

This equation can be solved for q, $q = -(H + \lambda I)^{-1} g$, and the result can be substituted into constraint (6c) to yield a scalar equation in the scalar unknown λ:

$$g^T (H + \lambda I)^{-2} g = 1 \tag{8}$$

If one multiplies both sides of eq. (8) by the square of the determinant of $(H + \lambda I)$, then the resulting equation is an 8th-order polynomial in λ, and the optimal λ can be determined by solving that polynomial. The global minimum of problem (6a)-(6c) occurs at the λ that is the largest (most positive) real solution of eq. (8) [14]. This λ value is the only real solution to eq. (8) that is greater than or equal to the negative of the minimum eigenvalue of H, which guarantees that the Hessian of the Lagrangian function, $H+\lambda I$, will at least be positive-semidefinite.

An efficient solution procedure for eq. (8) makes use of an eigenvalue decomposition of H:

$$H = V \begin{bmatrix} -\lambda_1 & 0 & 0 & 0 \\ 0 & -\lambda_2 & 0 & 0 \\ 0 & 0 & -\lambda_3 & 0 \\ 0 & 0 & 0 & -\lambda_4 \end{bmatrix} V^T \qquad \text{and} \qquad \begin{bmatrix} g_{z1} \\ g_{z2} \\ g_{z3} \\ g_{z4} \end{bmatrix} = V^T g \tag{9}$$

where $V = [v_1, v_2, v_3, v_4]$ is a matrix of orthogonal eigenvectors, $-\lambda_1 \geq -\lambda_2 \geq -\lambda_3 \geq -\lambda_4$ are the four eigenvalues, and g_{z1}, g_{z2}, g_{z3}, and g_{z4} are the components of a transformed gradient vector. Using the V transformation and the notation in eq. (9), eq. (8) can be rewritten in the following form:

$$f(\lambda) = \frac{g_{z1}^2}{(\lambda-\lambda_1)^2} + \frac{g_{z2}^2}{(\lambda-\lambda_2)^2} + \frac{g_{z3}^2}{(\lambda-\lambda_3)^2} + \frac{g_{z4}^2}{(\lambda-\lambda_4)^2} - 1 = 0 \tag{10}$$

Figure 1 shows a typical plot of $f(\lambda)$ vs. λ. The negatives of the eigenvalues of H are marked on the bottom of the plot as λ_1, λ_2, λ_3, and λ_4, and the optimal solution to eq. (8) is marked as λ_{opt}. Notice that it occurs at the highest value of λ for which $f(\lambda) = 0$. Notice, also, that there is an infinite peak in the plot at each of the λ_i values.

There is an efficient solution procedure that exploits the form of the curve in Fig. 1. It takes advantage of the facts that $\lambda_{opt} \geq \lambda_4$ and that $f(\lambda)$ is monotonically decreasing for $\lambda \geq \lambda_4$. These characteristics make it possible for any efficient numerical scalar equation solver, such as the guarded secant method, to determine λ_{opt} in very few iterations.

The solution of the original QUEST problem is a special case of this solution technique. As g_{z4} approaches 0, the $f(\lambda)$ spike at $\lambda = \lambda_4$ becomes infinitely narrow, and λ_{opt} will approach λ_4 if g_{z1}, g_{z2}, and g_{z3} are sufficiently small. All of the g_{zi} values are zero in the original QUEST algorithm; so, $\lambda_{opt} = \lambda_4$ in this case. When $\lambda_{opt} = \lambda_4$, the matrix $(H + \lambda I)$ is singular, but eq. (7) still has a solution. In fact, it has multiple solutions. An optimal solution is determined by inverting the nonsingular part of $(H + \lambda I)$ in eq. (7) to solve for the part of q that is a linear combination of the eigenvectors v_1, v_2, and v_3. The solution is completed by adding the term αv_4 to q and by selecting α to be large enough to satisfy the normality constraint in eq. (6c). There will be positive and negative values of α that satisfy the quaternion normalization, and both solutions will be global minima to problem (6a)-(6c) [14]. In the case of QUEST, the two q solutions, $\pm v_4$, are equivalent estimates of the attitude because $A(q) = A(-q)$.

EXTENDED QUEST FILTER ALGORITHM

Filtering Problem Statement

The extended QUEST filtering problem statement is defined for a single stage of a sampled data or discrete-time system. As will be shown, its solution leads to a natural method for iteration when dealing with multiple-stage systems. The problem is stated as a least-squares optimization problem, in keeping with the original Wahba formulation:

find: $q_{(k)}$ and $x_{(k)}$ {and $q_{(k-1)}$, $x_{(k-1)}$, and, $w_{(k-1)}$} (11a)

$$\text{to minimize:}\quad J = \frac{1}{2}\sum_{i=1}^{m_{(k)}} \frac{1}{\sigma_{i(k)}^2}\{b_{i(k)} - A[q_{(k)}]r_{i(k)}\}^T\{b_{i(k)} - A[q_{(k)}]r_{i(k)}\}$$

$$+ \frac{1}{2}\{R_{ww(k-1)}w_{(k-1)}\}^T\{R_{ww(k-1)}w_{(k-1)}\} + \frac{1}{2}\{R_{qq(k-1)}[q_{(k-1)} - \hat{q}_{(k-1)}]\}^T\{R_{qq(k-1)}[q_{(k-1)} - \hat{q}_{(k-1)}]\}$$

$$+ \frac{1}{2}\{R_{xq(k-1)}[q_{(k-1)} - \hat{q}_{(k-1)}] + R_{xx(k-1)}[x_{(k-1)} - \hat{x}_{(k-1)}]\}^T\{R_{xq(k-1)}[q_{(k-1)} - \hat{q}_{(k-1)}] + R_{xx(k-1)}[x_{(k-1)} - \hat{x}_{(k-1)}]\}$$

$$+ \, constant \tag{11b}$$

$$\text{subject to:}\quad q_{(k)} = \Phi\{t_{(k)}, t_{(k-1)}; q_{(k-1)}, x_{(k-1)}, w_{(k-1)}\}q_{(k-1)} \tag{11c}$$

$$x_{(k)} = f_x\{t_{(k)}, t_{(k-1)}; q_{(k-1)}, x_{(k-1)}, w_{(k-1)}\} \tag{11d}$$

$$q_{(k)}^T q_{(k)} = 1 \tag{11e}$$

The quantities in the above problem statement are defined as follows: q is the attitude quaternion, x is the vector of auxiliary filter states, and w is the process noise vector. The subscript $[\ ]_{(k)}$ refers to sample instant k, which occurs at time $t_{(k)}$, and the subscript $[\ ]_{(k-1)}$ refers to sample instant $k-1$, which occurs at time $t_{(k-1)}$ $(< t_{(k)})$. Just as in the usual QUEST cost function, the measured vectors $b_{i(k)}$ for $i = 1,...,m_{(k)}$ are the attitude reference unit direction vectors as measured in spacecraft coordinates at sample time $t_{(k)}$, the unit vectors $r_{i(k)}$ for $i = 1,...,m_{(k)}$ are the known inertial directions of the measured vectors, and the $\sigma_{i(k)}$ standard deviations are the per-axis accuracies of the $b_{i(k)}$ measurements. The vectors $\hat{q}_{(k-1)}$ and $\hat{x}_{(k-1)}$ are the *a posteriori* (or best) estimates of q and x at sample time $t_{(k-1)}$. The matrices $R_{ww(k-1)}$, $R_{qq(k-1)}$, $R_{xq(k-1)}$, and $R_{xx(k-1)}$ are weights that penalize the differences between $q_{(k-1)}$, $x_{(k-1)}$, and, $w_{(k-1)}$ and their *a posteriori* estimates at sample time $t_{(k-1)}$.

Equations (11c) and (11d) constitute the filter's dynamic model. The $4x4$ matrix $\Phi\{t_{(k)}, t_{(k-1)}; q_{(k-1)}, x_{(k-1)}, w_{(k-1)}\}$ is the orthogonal state transition matrix from time $t_{(k-1)}$ to time $t_{(k)}$ that is associated with the quaternion's kinematic differential equation:

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & \omega_3(t) & -\omega_2(t) & \omega_1(t) \\ -\omega_3(t) & 0 & \omega_1(t) & \omega_2(t) \\ \omega_2(t) & -\omega_1(t) & 0 & \omega_3(t) \\ -\omega_1(t) & -\omega_2(t) & -\omega_3(t) & 0 \end{bmatrix} q \tag{12}$$

In this equation $[\omega_1(t); \omega_2(t); \omega_3(t)] = \omega\{t; t_{(k-1)}, q_{(k-1)}, x_{(k-1)}, w_{(k-1)}\}$ is the attitude rate vector during the time interval $t_{(k-1)}$ to $t_{(k)}$. As shown in this formula, $\omega(t)$ may depend on the quaternion, the auxiliary state vector, or the process noise vector at time $t_{(k-1)}$. The specific form of this dependence will be dictated by the specific dynamic model that is used in the filter.

The remainder of the dynamic model consists of the discrete-time auxiliary state transition function $f_x\{t_{(k)}, t_{(k-1)}; q_{(k-1)}, x_{(k-1)}, w_{(k-1)}\}$. This is a vector function whose result has the same dimension as the auxiliary state vector x. This discrete-time function may be the result of numerical integration of auxiliary dynamic differential equations from time $t_{(k-1)}$ to time $t_{(k)}$, or it may be directly defined in discrete-time. The former situation holds if the x vector contains attitude rate estimates that get propagated via numerical integration of Euler's equations for an attitude dynamics model of the spacecraft. The latter situation may hold for a spacecraft that has rate gyros whose biases are estimated as part of x. If the auxiliary state vector is propagated between samples by numerical integration of a differential equation, then it will usually be best to integrate that equation and eq. (12) simultaneously.

It may be useful to pose a filter problem that has no measurements associated with some of the problem stages. In other words, $m_{(k)} = 0$ would hold for some (but not all) values of k. This might be needed because of the way in which process noise enters this model. It is modeled as a discrete-time process, $w_{(k-1)}$, $w_{(k)}$, $w_{(k+1)}$, ... One might want to use this discrete-time process noise to approximate the effects of a continuous-time white process noise. If the time period between actual attitude measurements is too large, then one needs to add extra "pseudo measurement" times between the actual attitude measurements. These extra "pseudo samples" break up the interval between measurements into multiple intervals over which different $w_{(k)}$

act. This modeling trick prevents the effective continuous-time process noise from losing its whiteness by having too large of a correlation time.

The problem in eqs. (11a)-(11e) is closely related to a combination of the square-root information filtering update and propagation problems of Ref. 15. Square-root information filters are normally developed by using least-squares estimation techniques, but they also admit a statistical interpretation. The statistical interpretation of the present filter is as follows: $R_{ww(k-1)}$ is the square root of the *a priori* information matrix for the random $w_{(k-1)}$ vector, and $[R_{qq(k-1)}, 0; R_{xq(k-1)}, R_{xx(k-1)}]$ is the square root of the *a posteriori* information matrix for the state estimate $[\hat{q}_{(k-1)}; \hat{x}_{(k-1)}]$.

The above problem form only admits vector attitude measurements, but it can easily be extended to include general attitude measurements. The attitude measurements appear in the first least-squares cost term on the right-hand side of eq. (11b). If other measurements were available, such as the cosine of the angle between a known inertial reference vector and a known spacecraft instrument vector, then an additional cost term would need to be added to the above problem statement to penalize the difference between the measured value of the cosine and the filter's best estimate of it. In that case, the filtering algorithm that will be presented below would need to linearize the resulting measurement equation. It would deal with such measurements just like any standard extended Kalman filter deals with such measurements. In this situation there would be no benefit from using the current filter beyond that of optimal maintenance of quaternion normalization, but there would also be no harm in using it.

New Filtering Algorithm

The new filtering algorithm is an iterative procedure that is made up of two phases per sample period. The algorithm that is developed here is an extended square-root information filtering algorithm [15], which forms its estimates by using least-squares-type procedures. Its first phase is a state and square-root information matrix propagation step. This phase starts with the *a posteriori* estimates at stage *k-1* and dynamically propagates them in order to compute the *a priori* estimates at stage *k*. This phase is carried out as in an extended Kalman filter, which means that the dynamics get linearized about stage *k-1*'s *a posteriori* state estimate in order to propagate the information matrices' square roots. The second phase of the process is the measurement update. This is the process of combining the *a priori* information at stage *k* with the vector attitude measurements at that stage in order to produce the best (or *a posteriori*) estimates of $q_{(k)}$ and $x_{(k)}$ given all of the available data up to that stage. This latter phase of the filtering process is where this paper makes its contributions.

Dynamic Propagation Phase

Although not new to this paper, the state propagation phase is presented here for purposes of completeness and in order to define notation for use later in the paper. The goal of propagation is to eliminate $q_{(k-1)}$, $x_{(k-1)}$, and $w_{(k-1)}$ from problem (11a)-(11e). The propagation procedure uses the constraints in eqs. (11c) and (11d) to eliminate $q_{(k-1)}$ and $x_{(k-1)}$. It next eliminates $w_{(k-1)}$ by partially optimizing the resulting cost function with respect to that quantity. This yields a cost function that depends only on $q_{(k)}$ and $x_{(k)}$. This cost is only an approximation of the original cost function because of the linearization of the dynamics about stage *k-1*'s *a posteriori* state estimate.

The propagation procedure first determines $\tilde{q}_{(k)}$ and $\tilde{x}_{(k)}$, the *a priori* estimates of $q_{(k)}$ and $x_{(k)}$. It does this by nonlinear propagation of eqs. (11c) and (11d):

$$\tilde{q}_{(k)} = \Phi\{t_{(k)},t_{(k-1)};\hat{q}_{(k-1)},\hat{x}_{(k-1)},0\}\hat{q}_{(k-1)} \tag{13a}$$

$$\tilde{x}_{(k)} = f_x\{t_{(k)},t_{(k-1)};\hat{q}_{(k-1)},\hat{x}_{(k-1)},0\} \tag{13b}$$

Note that $w_{(k-1)} = 0$ is used in this propagation because zero is its *a priori* expected value.

The filter next develops a linearized dynamic model. This model takes the form:

$$\begin{bmatrix} \Delta q_{(k)} \\ \Delta x_{(k)} \end{bmatrix} = \begin{bmatrix} \Phi_{qq(k-1)} & \Phi_{qx(k-1)} \\ \Phi_{xq(k-1)} & \Phi_{xx(k-1)} \end{bmatrix} \begin{bmatrix} \Delta q_{(k-1)} \\ \Delta x_{(k-1)} \end{bmatrix} + \begin{bmatrix} \Gamma_{q(k-1)} \\ \Gamma_{x(k-1)} \end{bmatrix} w_{(k-1)} \tag{14}$$

where the perturbations are defined to be $\Delta q_{(k)} = q_{(k)} - \tilde{q}_{(k)}$, $\Delta x_{(k)} = x_{(k)} - \tilde{x}_{(k)}$, $\Delta q_{(k-1)} = q_{(k-1)} - \hat{q}_{(k-1)}$, and $\Delta x_{(k-1)} = x_{(k-1)} - \hat{x}_{(k-1)}$. The state-transition and process noise effectiveness matrices in eq. (14) are Jacobians of the dynamics equations, (11c) and (11d):

$$\Phi_{qq(k-1)} = \Phi + \left[\frac{\partial \Phi}{\partial q}\right]\hat{q}_{(k-1)}, \qquad \Phi_{qx(k-1)} = \left[\frac{\partial \Phi}{\partial x}\right]\hat{q}_{(k-1)}, \qquad \Gamma_{q(k-1)} = \left[\frac{\partial \Phi}{\partial w}\right]\hat{q}_{(k-1)} \tag{15a}$$

$$\Phi_{xq(k-1)} = \frac{\partial f_x}{\partial q}, \qquad \Phi_{xx(k-1)} = \frac{\partial f_x}{\partial x}, \qquad \Gamma_{x(k-1)} = \frac{\partial f_x}{\partial w} \tag{15b}$$

with all of the partial derivatives evaluated at the point $[q_{(k-1)};x_{(k-1)};w_{(k-1)}] = [\hat{q}_{(k-1)};\hat{x}_{(k-1)};0]$.

The final operations of the propagation step are to form a large information matrix and to left QR-factorize it:

$$Q\begin{bmatrix} \tilde{R}_{qq(k)} & 0 & 0 \\ \tilde{R}_{xq(k)} & \tilde{R}_{xx(k)} & 0 \\ \tilde{R}_{wq(k)} & \tilde{R}_{wx(k)} & \tilde{R}_{ww(k-1)} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} R_{qq(k-1)} & 0 \\ R_{xq(k-1)} & R_{xx(k-1)} \end{bmatrix}\begin{bmatrix} \Phi_{qq(k-1)} & \Phi_{qx(k-1)} \\ \Phi_{xq(k-1)} & \Phi_{xx(k-1)} \end{bmatrix}^{-1}\begin{bmatrix} I & 0 & -\Gamma_{q(k-1)} \\ 0 & I & -\Gamma_{x(k-1)} \end{bmatrix} \\ \begin{bmatrix} 0 & \quad 0 & \quad R_{ww(k-1)} \end{bmatrix} \end{bmatrix} \tag{16}$$

where Q is an orthogonal matrix. The matrices $\tilde{R}_{qq(k)}$, $\tilde{R}_{xx(k)}$, and $\tilde{R}_{ww(k-1)}$ are square matrices, and $\tilde{R}_{xx(k)}$ and $\tilde{R}_{ww(k-1)}$ are both nonsingular. This orthogonal transformation can be carried out using techniques that are standard to square-root information filtering [15].

The result of the propagation step is a modified form of the cost function in eq. (11b):

$$\begin{aligned} J = {} & \frac{1}{2}\sum_{i=1}^{m_{(k)}} \frac{1}{\sigma_{i(k)}^2}\left\{b_{i(k)} - A[q_{(k)}]r_{i(k)}\right\}^T\left\{b_{i(k)} - A[q_{(k)}]r_{i(k)}\right\} + \frac{1}{2}\left\{\tilde{R}_{qq(k)}[q_{(k)} - \tilde{q}_{(k)}]\right\}^T\left\{\tilde{R}_{qq(k)}[q_{(k)} - \tilde{q}_{(k)}]\right\} \\ & + \frac{1}{2}\left\{\tilde{R}_{xq(k)}[q_{(k)} - \tilde{q}_{(k)}] + \tilde{R}_{xx(k)}[x_{(k)} - \tilde{x}_{(k)}]\right\}^T\left\{\tilde{R}_{xq(k)}[q_{(k)} - \tilde{q}_{(k)}] + \tilde{R}_{xx(k)}[x_{(k)} - \tilde{x}_{(k)}]\right\} \end{aligned} \tag{17}$$

This modified cost function is an approximation of the cost in eq. (11b) if the dynamic constraints in eqs. (11c) and (11d) are nonlinear, but it is exact if they are linear. Note that eq. (17) assumes that $w_{(k-1)}$ is set to its optimal value:

$$\{w_{(k-1)}\}_{opt} = -\tilde{R}_{ww(k-1)}^{-1}\left\{\tilde{R}_{wq(k)}[q_{(k)} - \tilde{q}_{(k)}] + \tilde{R}_{wx(k)}[x_{(k)} - \tilde{x}_{(k)}]\right\}. \tag{18}$$

Measurement Update Phase

The measurement update procedure solves a quadratically-constrained quadratic program. In order to form the quadratic program, it uses eqs. (3) and (4) to express the QUEST-type squared measurement-error cost terms as a quadratic form in $q_{(k)}$. Given that the resulting measurement error Hessian matrix is $H_{meas(k)}$, the measurement update problem becomes:

find: $q_{(k)}$ and $x_{(k)}$ (19a)

to minimize:
$$\begin{aligned} J = {} & \frac{1}{2}q_{(k)}^T H_{meas(k)} q_{(k)} + \frac{1}{2}\left\{\tilde{R}_{qq(k)}[q_{(k)} - \tilde{q}_{(k)}]\right\}^T\left\{\tilde{R}_{qq(k)}[q_{(k)} - \tilde{q}_{(k)}]\right\} \\ & + \frac{1}{2}\left\{\tilde{R}_{xq(k)}[q_{(k)} - \tilde{q}_{(k)}] + \tilde{R}_{xx(k)}[x_{(k)} - \tilde{x}_{(k)}]\right\}^T\left\{\tilde{R}_{xq(k)}[q_{(k)} - \tilde{q}_{(k)}] + \tilde{R}_{xx(k)}[x_{(k)} - \tilde{x}_{(k)}]\right\} + constant \end{aligned} \tag{19b}$$

subject to: $q_{(k)}^T q_{(k)} = 1$ (19c)

Optimization problem (19a)-(19c) can be solved in two steps, one that finds the optimum $x_{(k)}$ as a function of $q_{(k)}$ and the other that optimizes $q_{(k)}$. The $x_{(k)}$ auxiliary state vector does not enter constraint (19c). Therefore, it can be optimized by setting $\partial J/\partial x_{(k)}$ equal to zero and solving the resulting linear equation for $x_{(k)}$. The optimal value is

$$\{x_{(k)}\}_{opt} = \tilde{x}_{(k)} - \tilde{R}^{-1}_{xx(k)}\tilde{R}_{xq(k)}[q_{(k)} - \tilde{q}_{(k)}] \tag{20}$$

Substitution of the optimal $x_{(k)}$ from eq. (20) into the eq.-(19b) cost function yields the following least-squares optimization problem for determining the measurement update:

$$\text{find:} \quad q_{(k)} \tag{21a}$$

$$\text{to minimize:} \quad J = \tfrac{1}{2}q^T_{(k)}H_{meas(k)}q_{(k)} + \tfrac{1}{2}\{\tilde{R}_{qq(k)}[q_{(k)} - \tilde{q}_{(k)}]\}^T\{\tilde{R}_{qq(k)}[q_{(k)} - \tilde{q}_{(k)}]\} + constant \tag{21b}$$

$$\text{subject to:} \quad q^T_{(k)}q_{(k)} = 1 \tag{21c}$$

This problem is in the same general form as the quadratically-constrained quadratic program in eqs. (6a)-(6c). This equivalence is evident if one defines: $H = H_{meas(k)} + \tilde{R}^T_{qq(k)}\tilde{R}_{qq(k)}$ and $g = -\tilde{R}^T_{qq(k)}\tilde{R}_{qq(k)}\tilde{q}_{(k)}$.

The measurement update is completed by solving problem (6a)-(6c) with the H matrix and g vector given above. The solution to this problem is the *a posteriori* estimate of the attitude quaternion at sample k, $\hat{q}_{(k)}$. This value of $q_{(k)}$ is next substituted into eq. (20) to compute the *a posteriori* auxiliary state vector estimate at sample k, $\hat{x}_{(k)} = \tilde{x}_{(k)} - \tilde{R}^{-1}_{xx(k)}\tilde{R}_{xq(k)}[\hat{q}_{(k)} - \tilde{q}_{(k)}]$.

In order for the algorithm to be recursive, the measurement update procedure must re-express the cost function in eq. (19b) in terms like those of eq. (11b) that constitute the *a posteriori* cost function at sample $k-1$ – the 3rd and 4th terms on the right-hand side of eq. (11b). This cost function must be a quadratic form in $[q_{(k)} - \hat{q}_{(k)}]$ and $[x_{(k)} - \hat{x}_{(k)}]$.

It is possible to derive a cost function in the required form that is equivalent to the cost in eq. (19b) on the manifold $q^T_{(k)}q_{(k)} = 1$:

$$\begin{aligned} J = {} & \tfrac{1}{2}[q_{(k)} - \hat{q}_{(k)}]^T\{H_{meas(k)} + \tilde{R}^T_{qq(k)}\tilde{R}_{qq(k)} + \lambda_{(k)}I\}[q_{(k)} - \hat{q}_{(k)}] \\ & + \tfrac{1}{2}\{\tilde{R}_{xq(k)}[q_{(k)} - \hat{q}_{(k)}] + \tilde{R}_{xx(k)}[x_{(k)} - \hat{x}_{(k)}]\}^T\{\tilde{R}_{xq(k)}[q_{(k)} - \hat{q}_{(k)}] + \tilde{R}_{xx(k)}[x_{(k)} - \hat{x}_{(k)}]\} + constant \end{aligned} \tag{22}$$

Note that $\lambda_{(k)}$ in this equation is two times the optimal Lagrange multiplier for the quaternion normalization constraint. Equivalence between this cost function and the eq.-(19b) cost function on the manifold is sufficient for the purpose of recursive filtering because all optimizations at successive sample instants are effectively constrained to this manifold, at least to first order in the perturbations $[q_{(k)} - \hat{q}_{(k)}]$ and $[x_{(k)} - \hat{x}_{(k)}]$.

The cost function in eq. (22) can be derived by making use of the optimality necessary conditions for the measurement update,

$$\tilde{R}^T_{xx(k)}\{\tilde{R}_{xq(k)}[\hat{q}_{(k)} - \tilde{q}_{(k)}] + \tilde{R}_{xx(k)}[\hat{x}_{(k)} - \tilde{x}_{(k)}]\} = 0 \tag{23a}$$

$$\{H_{meas(k)} + \tilde{R}^T_{qq(k)}\tilde{R}_{qq(k)} + \lambda_{(k)}I\}\hat{q}_{(k)} - \tilde{R}^T_{qq(k)}\tilde{R}_{qq(k)}\tilde{q}_{(k)} = 0 \tag{23b}$$

and the following condition

$$\tfrac{1}{2}[q_{(k)} - \hat{q}_{(k)}]^T[q_{(k)} - \hat{q}_{(k)}] = -\hat{q}^T_{(k)}q_{(k)} + 1 \tag{24}$$

which can be derived from the quaternion normalization constraint. The detailed derivation of eq. (22) is straightforward, but it has been omitted for the sake of brevity.

One last step completes the preparations for recursive application of the algorithm at an incremented value of k. This step is to express the eq.-(22) cost as a sum of squares:

$$J = + \tfrac{1}{2}\{R_{qq(k)}[q_{(k)} - \hat{q}_{(k)}]\}^T\{R_{qq(k)}[q_{(k)} - \hat{q}_{(k)}]\}$$
$$+ \tfrac{1}{2}\{R_{xq(k)}[q_{(k)} - \hat{q}_{(k)}] + R_{xx(k)}[x_{(k)} - \hat{x}_{(k)}]\}^T \{R_{xq(k)}[q_{(k)} - \hat{q}_{(k)}] + R_{xx(k)}[x_{(k)} - \hat{x}_{(k)}]\} + constant \qquad (25)$$

where $R_{qq(k)}$ is a matrix square root: $R^T_{qq(k)}R_{qq(k)} = \{H_{meas(k)} + \tilde{R}^T_{qq(k)}\tilde{R}_{qq(k)} + \lambda_{(k)}I\}$ and where $R_{xq(k)} = \tilde{R}_{xq(k)}$ and $R_{xx(k)} = \tilde{R}_{xx(k)}$. The matrix square root $R_{qq(k)}$ is guaranteed to exist because the matrix $\{H_{meas(k)} + \tilde{R}^T_{qq(k)}\tilde{R}_{qq(k)} + \lambda_{(k)}I\}$ is at least positive-semidefinite at the global minimum of problem (21a)-(21c), and it may be positive definite. This matrix square root can be computed using the eigenvalue/eigenvector decomposition of $\{H_{meas(k)} + \tilde{R}^T_{qq(k)}\tilde{R}_{qq(k)}\}$ which is part of the solution procedure for a quadratically-constrained quadratic program that is presented in the second section of this paper.

Comparison with other QUEST Algorithms.

The present algorithm is a generalization of the QUEST algorithms presented in Refs 9 and 10. It reduces exactly to the previous algorithms under appropriate problem modeling assumptions. First of all, if no dynamics are assumed in the problem and if there is no auxiliary state vector to estimate, then only the first term on the right-hand side of eq. (11b) remains in the cost function, and constraints (11c) and (11d) disappear. This yields a problem that is exactly Wahba's original problem [11]. The measurement update part of the filter will calculate a $q_{(k)}$ estimate that is the appropriate eigenvector of the cost function's Hessian matrix. This will happen because $g = 0$ will hold in the general quadratically-constrained quadratic program, and its solution algorithm will recognize this as a special case in which it must make use of eigenvectors.

Alternate modeling assumptions yield the recursive QUEST algorithms of Ref. 10. That paper presents two recursive algorithms, one for use with perfect rate-gyro measurements and one for use with noisy rate gyros whose noise is approximately modeled as white noise. In both cases these filters can be re-derived by using problem models that have quaternion dynamics as in eq. (11c), but that have no auxiliary state vector.

In the perfect rate-gyro case, one can show that the cost in eq. (19b) is equivalent to

$$J = \tfrac{1}{2}q^T_{(k)}H_{meas(k)}q_{(k)} + \tfrac{1}{2}[\Phi^T_{qq(k-1)}q_{(k)} - \hat{q}_{(k-1)}]^T\{H_{(k-1)} + \lambda_{(k-1)}I\}[\Phi^T_{qq(k-1)}q_{(k)} - \hat{q}_{(k-1)}] + constant \qquad (26)$$

where $H_{(k-1)}$ is the filter cost function's cumulative Hessian matrix at sample $k-1$. Equation (26) holds because $\Phi_{qq(k-1)}$ equals the orthogonal Φ matrix of eq. (11c) in this situation and because there is no $w_{(k-1)}$ process noise. The optimality necessary condition at sample $k-1$, $\{H_{(k-1)} + \lambda_{(k-1)}I\}\hat{q}_{(k-1)} = 0$, can be used to reduce eq. (26) to the following form

$$J = \tfrac{1}{2}q^T_{(k)}\{H_{meas(k)} + \Phi_{qq(k-1)}H_{(k-1)}\Phi^T_{qq(k-1)}\}q_{(k)} + \tfrac{1}{2}\lambda_{(k-1)} + constant \qquad (27)$$

Note that the bracketed expression in this equation constitutes $H_{(k)}$.

This cost form proves that the present algorithm is equivalent to the first recursive algorithm of Ref. 10 under the stated modeling assumptions. Except for a scale factor, the result in eq. (27) is equivalent to the cost that would be calculated by the algorithm of Ref. 10. The present filter estimates $q_{(k)}$ by minimizing the cost in eq. (26) subject to the normalization constraint. Reference 10's first algorithm minimizes the cost in eq. (27) and, therefore, calculates the same quaternion estimate as does the present algorithm.

The second recursive QUEST algorithm of Ref. 10 can be reproduced by the present filter with slightly different modeling assumptions. Reference 10's second filter has a forgetting factor, $\rho_{(k-1)}$. It is a number between 0 and 1 and is used to de-weight the cumulative measurement error cost terms up to sample time $k-1$ before adding them to the new measurement error cost terms at sample time k.

This same forgetting factor effect can be reproduced in the present filter by adding an appropriately modeled process noise, $w_{(k-1)}$. In this model, one assumes rate-gyro measurements that are numerically integrated to determine $\Phi_{qq(k-1)}$, which again equals the orthogonal Φ matrix of eq. (11c). The discrete-time, white-noise rate-gyro measurement error is a 3-dimensional process noise vector that can produce rotation errors in 3 orthogonal directions. The linearized influence of $w_{(k-1)}$ on the dynamics model is characterized by the 4×3 matrix $\Gamma_{q(k-1)}$. The 3 columns of this matrix are orthogonal to each other and to $\Phi\ \hat{q}_{(k-1)}$. If the model also assumes that the process disturbance noise information matrix is

$$R^T_{ww(k-1)}R_{ww(k-1)} = \left\{\frac{\rho_{(k-1)}}{1-\rho_{(k-1)}}\right\}\Gamma^T_{q(k-1)}\Phi_{qq(k-1)}\left\{H_{(k-1)} + \lambda_{(k-1)}I\right\}\Phi^T_{qq(k-1)}\Gamma_{q(k-1)} \tag{28}$$

then the cost function in eq. (19b) can be reduced to the form

$$J = \tfrac{1}{2}q^T_{(k)}\left\{H_{meas(k)} + \rho_{(k-1)}\Phi_{qq(k-1)}H_{(k-1)}\Phi^T_{qq(k-1)}\right\}q_{(k)} + constant \tag{29}$$

Except for a scale factor, this is the same sample-*k* cost as is used in the 2nd filter of Ref. 10, which proves that this implementation of the present filter is equivalent to it.

THE ESTIMATION ERROR COVARIANCE

There is a direct connection between the square-root information filter that has been developed and the covariance of the optimal estimate. It is well known that there is such a connection for standard square-root information filters [15]. Three issues must be dealt with in order to generalize the standard results to the present case. One is system nonlinearity, another is the statistical model of the QUEST measurement errors, and the third is the quaternion normalization constraint.

The issue of nonlinearity will be dealt with in the usual way for extended Kalman filters. It will be assumed that accurate calculations can be made using linearizations around *a priori* and *a posteriori* estimates. This assumption will hold so long as the measurement noise and process noise are not "large" compared to the nonlinear terms according to some sensible definition of "large."

The measurement errors can be modeled statistically by means of the following probability density function for the measurements $b_{1(k)}, \ldots, b_{m(k)}$ conditioned on the quaternion, $q_{(k)}$:

$$\mathcal{P}[b_{1(k)}, \ldots, b_{m(k)}|\ q_{(k)}] = C\ exp\{-J_{QUEST(k)}[q_{(k)}]\} \tag{30}$$

where C is a constant. This function defines probability density on the manifold $q^T_{(k)}q_{(k)} = 1$. This probability density function is sensible. For each measured b vector this function gives an error probability density function of $C\ exp\{-2sin^2(\theta/2)/\sigma^2\}$ where θ is the angle between the measured b vector and its true direction and where σ is the angular error's standard deviation. This density function approaches a Gaussian for small σ.

The constraint that the estimate lie on the manifold $q^T_{(k)}q_{(k)} = 1$ is handled in the following way. First, one recognizes that this constraint causes the covariance matrix to be singular, with its null space being the normal to the constraint manifold. Expressed in plain terminology, the estimator knows the quaternion length exactly; so, the variance of the quaternion length is zero. The remainder of the covariance calculations are carried out in the manifold's local tangent space. Although not valid for large uncertainties, this approach is consistent with the linearizing assumptions that are used to deal with all other problem nonlinearities for purposes of calculating covariances.

The rest of the covariance calculation proceeds in a manner analogous to standard square-root information filter theory. One assumes that the *a priori* process noise covariance matrix is $\{R^T_{ww(k-1)}R_{ww(k-1)}\}^{-1}$. The *a posteriori* estimation error covariance at sample k is then the inverse of the projection of the matrix $[R_{qq(k)}, 0; R_{xq(k)}, R_{xx(k)}]^T[R_{qq(k)}, 0; R_{xq(k)}, R_{xx(k)}]$ onto the subspace that is tangent to the quaternion normalization constraint. This covariance is calculated as follows. First, one uses left QR factorization to determine 3 quaternion vectors that are mutually orthogonal to each other and to $\hat{q}_{(k)}$:

$$[\hat{q}_{(k)}, Q_2] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = Q_{ns} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \hat{q}_{(k)} \tag{31}$$

where Q_{ns} is an orthogonal matrix. The 4×3 matrix Q_2, which forms the last 3 columns of Q_{ns}, is used to perform the projections that are needed in order to calculate the covariance matrix:

$$\begin{bmatrix} \hat{P}_{qq(k)} & \hat{P}_{qx(k)} \\ \hat{P}^T_{qx(k)} & \hat{P}_{xx(k)} \end{bmatrix} = \begin{bmatrix} Q_2 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} Q_2^T (R^T_{qq(k)} R_{qq(k)} + R^T_{xq(k)} R_{xq(k)}) Q_2 & Q_2^T R^T_{xq(k)} R_{xx(k)} \\ R^T_{xx(k)} R_{xq(k)} Q_2 & R^T_{xx(k)} R_{xx(k)} \end{bmatrix}^{-1} \begin{bmatrix} Q_2^T & 0 \\ 0 & I \end{bmatrix} \tag{32}$$

It is straightforward to show that this covariance matrix is singular, having its one zero eigenvalue along the eigenvector direction $[\hat{q}_{(k)};0]$.

It can be proved rigorously that the filter's *a posteriori* estimates $\hat{q}_{(k)}$ and $\hat{x}_{(k)}$ are the expected values of $q_{(k)}$ and $x_{(k)}$ conditioned on the measurements up through sample k and on the *a priori* statistics. It can also be proved that the formula in eq. (32) gives the correct covariance for the errors in these optimal estimates. The proofs make use of the linearization assumptions; so, they are valid only for "small" uncertainties. The proofs work with various conditional probability density functions that define probabilities on manifolds of the form $q^T q = 1$. These probability density functions all take the form $\rho(q,x) = C \exp\{-J(q,x)\}$, where $J(q,x)$ is one of the least-squares cost functions defined above. Although straightforward, the proofs are lengthy. They have been omitted for the sake of brevity.

SIMULATION TESTS OF THE ALGORITHM

The algorithm has been tested using simulated data. There are several reasons for simulation testing. One is to check for any unforeseen difficulties with implementation of the algorithm. Another is to determine whether the algorithm indeed has better convergence properties than a standard extended Kalman filter. Yet a third reason for testing is to check out the practicality of the small-angle assumptions in the covariance analysis.

A baseline extended Kalman filter has also been tested in order to determine whether the new filter has an improved ability to converge. The baseline filter is almost the same as the above filter except that it linearizes the measurement error equations about the *a priori* quaternion estimate before it calculates the measurement error. This leads to a measurement error cost function of the following form:

$$J_{meas}(q) = \frac{1}{2} \sum_{i=1}^{m} \frac{1}{\sigma_i^2} \{ b_i - [A(\tilde{q}) + \frac{\partial A}{\partial q}(q - \tilde{q})] r_i \}^T \{ b_i - [A(\tilde{q}) + \frac{\partial A}{\partial q}(q - \tilde{q})] r_i \} \tag{33}$$

This is exactly what a standard extended Kalman filter does with measurement errors if it is implemented as a square-root information filter. This baseline filter uses the quadratically-constrained measurement update, which optimally enforces quaternion normalization. Therefore, it is slightly more sophisticated than the extended Kalman filters of Refs. 4, 5, and 7.

Filter Design

A relatively simple filtering case has been tested. It assumes the availability of noisy rate-gyro data and star-tracker data. The rate gyro is assumed to have biases. The filter's estimation vector is $[q; \omega_{bias}]$, where ω_{bias} is the estimated rate-gyro bias vector. It constitutes this filter's auxiliary state vector, x. The dynamic model of the filter consists of a model for the angular velocity vector between measurement samples and a model for the rate-gyro bias dynamics. The angular velocity model that gets used in eq. (12) is:

$$\omega\{t; t_{(k-1)}, \hat{q}_{(k-1)}, \hat{\omega}_{bias(k-1)}, w_{(k-1)}\} = \omega_{rg(k-1)} + \left(\frac{t - t_{(k-1)}}{t_{(k)} - t_{(k-1)}} \right) [\omega_{rg(k)} - \omega_{rg(k-1)}] - \hat{\omega}_{bias(k-1)} - w_{a(k-1)} \tag{34}$$

where $\omega_{g(k)}$ is the rate-gyro measurement at sample k and $w_{a(k-1)}$ constitutes the first 3 elements of the 6×1 process noise vector $w_{(k-1)}$. The dynamic model of the rate-gyro bias vector is a random walk:

$$\omega_{bias(k)} = \omega_{bias(k-1)} + w_{b(k-1)} \tag{35}$$

where $w_{b(k-1)}$ constitutes the last 3 elements of the $w_{(k-1)}$ process noise vector; i.e., $w_{(k-1)} = [w_{a(k-1)}; w_{b(k-1)}]$.

Truth Model

This filter has been tested with data from a simulated truth model. The simulated truth model is that of a rigid-body spacecraft in a low Earth orbit. The truth model simulates Euler's equations and the quaternion kinematics. It includes gravity-gradient torques and white-noise disturbance torques. One scenario tests a spin-stabilized spacecraft that undergoes nutations. It has a spin period of *50* sec and a nutation period of *504* sec. The other scenario tests a nadir-pointing gravity-gradient stabilized spacecraft that librates at frequencies on the order of the orbital frequency.

The truth model includes a model of the star tracker and of the rate gyro. The star tracker is assumed to have a limited field of view; it has only a *5°* radius. For the spin-stabilized spacecraft, the center of the star-tracker's field of view points approximately perpendicular to the nominal spin vector. For the nadir-pointing spacecraft it points towards the nominal zenith direction. Similar to what was done in Ref. 13, star-tracker measurements have been simulated by randomly generating a direction vector in the star tracker's field of view at each measurement sample. That direction and the spacecraft's true attitude have been used to generate the "known" inertial direction vector for that attitude measurement, $r_{i(k)}$. The $b_{i(k)}$ measured vector has been calculated by taking the original randomly generated direction vector and adding a random direction error component that has a Gaussian distribution with a standard deviation of *10* arc sec per axis. In the case of the spin stabilized spacecraft, the star tracker makes one measurement every 15 seconds, which is about 3 times per spin period. For the nadir-pointing spacecraft, the star tracker measures the direction to one star once every 58.5 seconds, or 100 times per orbit.

The measurement model for the rate gyro includes errors due to white noise and errors due to a bias that can drift as a random-walk. Two different intensities have been used for the white-noise component of the rate gyro error: *0.02°*/hour$^{1/2}$ and *0.10°*/hour$^{1/2}$. The intensity of the white noise that drives the bias drift has been set at (*0.10°*/hour)/hour$^{1/2}$ for all cases in this paper.

It is important to choose small enough sample rates for the rate gyros. The sample rates used in this study are once every *0.625* sec for the spin-stabilized spacecraft and once every *2.92* sec for the nadir-pointing spacecraft. Lower sampling rates can cause systematic estimation errors due to the truncation error that is inherent in the filter's angular rate model, i.e., in eq. (34).

There are sample times with no measurement update. This happens because of differences between the star-tracker sampling rates and the rate-gyro sampling rates. In the spinning spacecraft case there is one star-tracker measurement for every *24* rate-gyro measurements. In this case, $t_{(k)} - t_{(k-1)}$ is fixed at *0.625* sec, but $m_{(k)}$ varies. It is *0* for *23* out of *24* samples, and then it is *1* for the *24th* sample. The nadir-pointing spacecraft has a $t_{(k)} - t_{(k-1)}$ of *2.92* sec, and $m_{(k)}$ switches from *0* to *1* every *20th* sample.

Results

This filter shows very good performance under nominal conditions and when subjected to large initial attitude uncertainty. As an example, Fig. 2 shows attitude error results for a filtering case that used simulated data for the spinning spacecraft. This case starts with a moderate initial attitude error; the error is *11°* in magnitude and directed about an axis that is *43°* away from the nominal spin axis. Its initial rate-gyro bias error is small, with a magnitude of *1.4°*/hour, and its rate-gyro error's white-noise intensity is *0.02°*/hour$^{1/2}$. Figure 2 shows the total attitude error – the total rotational error between the *a posteriori* attitude estimate and the true attitude. It also shows the filter's predicted *a posteriori* standard deviation for this quantity. It is clear from this plot that the filter does a good job of attitude determination and that its covariance calculations are consistent with the actual errors.

A number of cases have been run in order to test the new filter's ability to converge from large initial attitude errors. In each of these cases, the baseline extended Kalman filter has been used to filter the same data. This provides a point of comparison that allows one to determine whether the new filter can converge in situations where the baseline filter cannot. Figures 3 and 4 present results for one of these cases, a case that considers the nadir-pointing spacecraft. Both filters were given the same large initial errors in the attitude and in the rate-gyro bias estimates. The initial attitude error was *180°* about the spacecraft's roll axis, and the initial rate-gyro bias error was *100°*/hour about the pitch axis. To make matters even worse, the filter was given erroneous standard deviations for its *a priori* per-axis attitude errors and rate-gyro bias errors, *0.1°* and *1°*/hour, respectively. The rate-gyro's white-noise error component had an intensity of *0.10°*/hour$^{1/2}$ for this case.

The baseline extended Kalman filter fails to converge from this poor first guess, but the extended QUEST attitude filter does very well. The extended QUEST filter converges to an attitude error of less than *3°* in its first *500* sec of filtering, and its rate-gyro bias error gets reduced to a magnitude of under *3°*/hour after one orbit of filtering.

Even with substantial decreases in the initial errors, the standard extended Kalman filter still has problems. If the initial attitude and rate-gyro bias errors are reduced by *25%* from those used in Figs. 3 and 4, then the standard filter still fails to converge. With a *33.3%* reduction – to an initial attitude error of *120°* – the standard filter finally converges, but its attitude error is still *3.8°* after one orbit. When the extended QUEST filter is used for this same *120°*-case, it reduces the error to less than *3.8°* right at the outset and to about *0.1°* in less than half an orbit.

In almost all large-initial-error cases considered, the extended QUEST filter displayed better performance than the extended Kalman filter. In many cases the extended Kalman filter failed to converge when the extended QUEST filter succeeded. There was one case where the extended Kalman filter converged and the new filter failed to converge, but this was a spurious case in which the extended Kalman filter converged only by accident. Even when the extended Kalman filter did converge, it almost always took much more time to achieve a good estimate than did the new extended QUEST filter.

Another difference between the two filters is in their sensitivities to tuning when initial errors are large. The extended Kalman filter can be made to diverge or converge by changing the initial *a priori* state error covariance or by changing the rate gyro's white-noise error intensity. In at least one case, the relationship between convergence and the covariance tuning proved to be counter-intuitive. The extended QUEST filter, on the other hand, exhibits insensitivity to changes in covariance tuning.

It must be noted that a poor initial guess can have detrimental effects on the new extended QUEST attitude filter, especially if the initial filter covariances are not set properly. This can be seen if one looks at the filter's predicted estimation error standard deviations for the case that produced Figs. 3 and 4. The initial attitude error is *1,800* times larger than the initial standard deviation for this quantity. The filter does not get the actual attitude error to be less than *10* times the filter's *a posteriori* standard deviation until after *4,500* sec of filtering. The rate-gyro bias behaves the same way: for the whole first orbit, the actual rate-gyro errors remain more than an order of magnitude larger than the filter's *a posteriori* standard deviation. If the measurement and process-noise covariances in the filter were close to the real system's actual values, then this problem would go away after a long time. In the interim, before the covariances settled down to their correct values, this discrepancy would cause the actual errors to be larger than they would have been with better tuning.

Even though the new filter has an increased ability to converge from initial attitude errors, it still can fail to converge if the error in its initial estimate of the auxiliary state vector is too large. With *180°* initial attitude errors for the nadir-pointing spacecraft, as in Figs. 3 and 4, the filter successfully converged from an initial rate-gyro bias error magnitude of *2,400°*/hour, but an initial error of *4,000°*/hour caused divergence. At a *3,200°*/hour initial error, the filter did not converge after one orbit, and it was unclear whether it would ever converge. Similar behavior has been found for the spinning spacecraft, where divergence has been observed for initial rate-gyro bias errors above *12,000°*/hour.

This demonstrates that the filter may be unable to converge from large initial errors in its auxiliary state vector, x. Any failure to converge is due to the linearization assumption that was made in eq. (14). Probably the only way to ensure that a filter cannot diverge is to do batch filtering with numerical iteration, as in Ref. 12.

The new filter's convergence properties are probably sufficiently robust for almost any mission. Even though the filter can diverge for a wrong initial guess of x, it appears to have a very large domain of convergence in the examples that have been considered. In the nadir-pointing case, convergence has been achieved for an initial rate-gyro bias error that is more than 10 times as large as the orbital rate. In the spinning spacecraft case, the filter successfully converged even when the initial rate-gyro bias error was 48% of the spacecraft's spin rate.

The extended QUEST attitude determination filter is capable of working with a system model that propagates its estimate of the attitude rate vector using Euler's equations. None of the examples included in this section use such a model because of limits to the scope of this effort. If Euler's equations are used, then filter convergence may depend on not having too large of an initial error in q, contrary to what has been found in the above examples. This could happen because of q's effect on gravity-gradient torques or on other terms in Euler's equations. Although the filter might be more prone to diverge, it still might out-perform a standard extended Kalman filter in this regard. Of course, more work is needed in order to investigate this issue.

CONCLUSIONS

A new spacecraft attitude determination filter has been developed. It operates on vector attitude data. Its goal is to incorporate QUEST-type measurement updates into an extended Kalman filter framework in hopes of improving the filter's convergence robustness in the face of large initial attitude errors and nonlinear effects. The extended filter's state vector uses a quaternion attitude parameterization and can include other elements such as angular rates or rate-gyro biases. The filter uses standard square-root information filtering techniques wherever possible. The only exception is that the quaternion part of the measurement update involves the solution of a quadratically-constrained quadratic program, as in the original QUEST algorithm. The quadratic cost function gets modified in this case to include the effects of *a priori* information and the effects of the other state vector elements.

The new algorithm has proved successful at increasing the range of initial attitude uncertainties from which the filter can converge to the true attitude. In an example that involved star-tracker and rate-gyro measurements with rate-gyro bias estimation, the extended QUEST algorithm was able to converge from simultaneous initial errors of *180°* in attitude and *2,400°*/hour in rate-gyro bias. Larger initial rate-gyro bias errors can cause divergence, but for the cases considered, the size of the initial attitude error has no effect on the filter's ability to converge. Such properties will be important to spacecraft missions that require an increased degree of autonomy and, therefore, an increased domain of convergence for the attitude filter.

REFERENCES

1. Wertz, J.R. ed., *Spacecraft Attitude Determination and Control*, D. Reidel Pub. Co., (Boston, 1978).

2. Cohen, C.E., "Attitude Determination," in *Global Positioning System: Theory and Applications, Vol. II*, Parkinson, B.W. and Spilker, J.J. Jr., eds., American Institute of Aeronautics and Astronautics, (Washington, 1996), pp. 519-538.

3. Lefferts, E.J., Markley, F.L., and Shuster, M.D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, 1982, pp. 417-429.

4. Bar-Itzhack, I.Y. and Oshman, Y., "Attitude Determination from Vector Observations: Quaternion Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-21, No. 1, 1985, pp. 128-136.

5. Shuster, M.D., "Kalman Filtering of Spacecraft Attitude and the QUEST Model," *Journal of the Astronautical Sciences*, Vol. 38, No. 3, 1990, pp. 377-393.

6. Psiaki, M.L., Martel, F., and Pal, P.K., "Three-Axis Attitude Determination via Kalman Filtering of Magnetometer Data," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 506-514.

7. Bar-Itzhack, I.Y., Deutschmann, J., and Markley, F.L., "Quaternion Normalization in Additive EKF for Spacecraft Attitude Determination," *Proceedings of the AIAA Guidance, Navigation, and Control Conf.*, New Orleans, LA, Aug. 1991, pp. 908-916.

8. Psiaki, M.L., Theiler, J., Bloch, J., Ryan, S., Dill, R.W., and Warner, R.E. "ALEXIS Spacecraft Attitude Reconstruction with Thermal/Flexible Motions Due to Launch Damage", *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, Sept.-Oct. 1997, pp. 1033-1041.

9. Shuster, M.D. and Oh, S.D., "Three-Axis Attitude Determination from Vector Observations," *Journal of Guidance and Control*, Vol. 4, No. 1, 1981, pp. 70-77.

10. Bar-Itzhack, I.Y., "REQUEST: A Recursive QUEST Algorithm for Sequential Attitude Determination," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, Sept.-Oct. 1996, pp. 1034-1038.

11. Wahba, G., "A Least Squares Estimate of Satellite Attitude," *SIAM Review*, Vol. 7, No. 3, 1965, p. 409.

12. Markley, F.L., "Attitude Determination and Parameter Estimation Using Vector Observations: Theory," *Journal of the Astronautical Sciences*, Vol. 37, No. 1, 1989, pp. 41-58.

13. Markley, F.L., "Attitude Determination and Parameter Estimation Using Vector Observations: Application," *Journal of the Astronautical Sciences*, Vol. 39, No. 3, 1991, pp. 367-381.

14. Sorensen, D.C., "Newton's Method with a Model Trust Region Modification," *SIAM Journal on Numerical Analysis*, Vol. 19, No. 2, 1982, pp. 409-426.

15. Bierman, G.J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, (New York, 1977), pp. 57-67, 69-76, and 115-122.

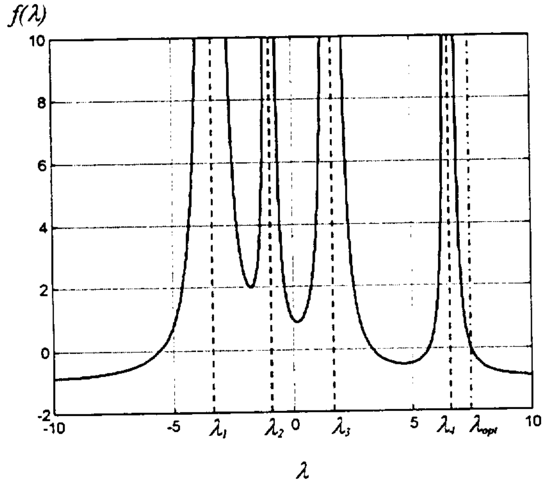


Fig. 1. Constraint function dependence on the Lagrange multiplier in the solution of the quadratically-constrained quadratic program.

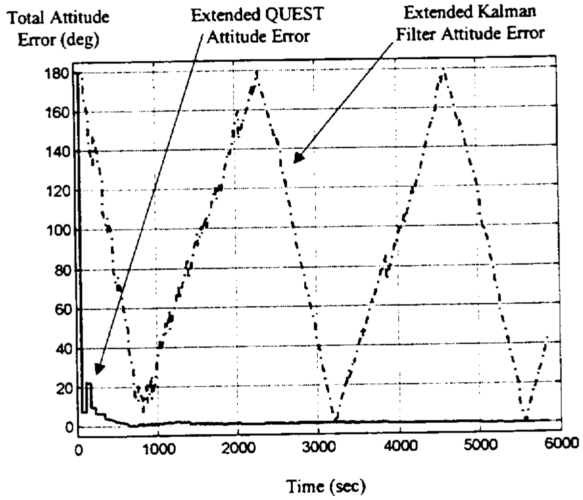


Fig. 3. Comparison of total attitude error time histories for 2 filters that start with a poor a priori state estimate.

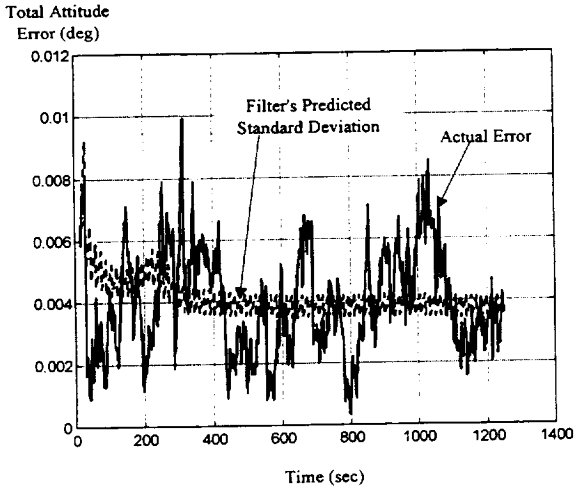


Fig. 2. Total attitude error time history and its predicted standard deviation for the extended QUEST filter operating on data from the spinning spacecraft.

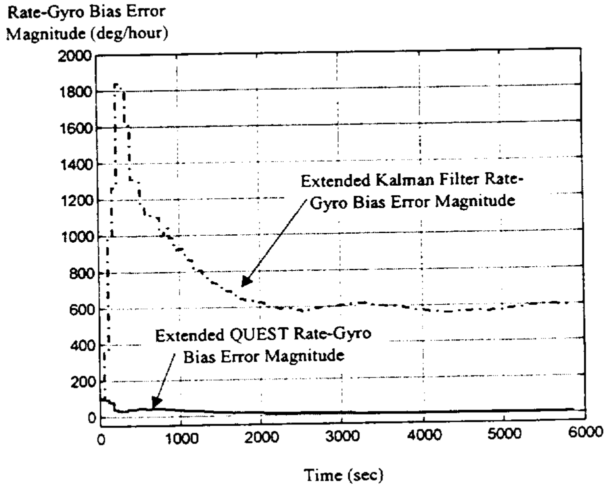


Fig. 4. Comparison of rate-gyro bias error magnitude time histories for 2 filters that start with a poor a priori state estimate.

ASCAL: AUTONOMOUS ATTITUDE SENSOR CALIBRATION

Chariya Peterson and John Rowe
Computer Sciences Corp. 10110 Aerospace Rd. Lanham-Seabrook, MD 20706

Karl Mueller and Nigel Ziyad
NASA Goddard Space Flight Center, Greenbelt, MD 20771

Abstract

In this paper, an approach to increase the degree of autonomy of flight software is proposed. We describe an enhancement of the Attitude Determination and Control System by augmenting it with self-calibration capability. Conventional attitude estimation and control algorithms are combined with higher level decision making and machine learning algorithms in order to deal with the uncertainty and complexity of the problem.

1 Introduction

The goal of our project is to enhance the degree of autonomy of the Attitude Determination and Control System (ADCS), enabling it to perform accurately without human intervention for an extended period of time. The approach is to evolve ADCS one step at a time into an autonomous system in a natural way dictated by actual needs. The purpose of this paper is to describe the first step in our program: the development of the Autonomous Attitude Sensor Calibration (ASCAL). The intention is to demonstrate ideas and concepts of on-board autonomy evolving from the existing control system, and not to develop another technique of attitude sensor calibration. A conventional ADCS uses data from available attitude sensors to estimate the attitude of the spacecraft. To meet mission pointing accuracy requirements, the attitude sensors must be calibrated for instrument biases, scale factors and misalignments immediately after launch and as needed thereafter. Traditionally, the calibration process is performed by attitude support specialists, often requiring elaborate procedures involving attitude consistency checks, data sampling and trending, and diagnosis expertise. A system that is able to perform all of these functions autonomously will have to deal with a large degree of uncertainty due to errors in the model parameters, incomplete model, measurement errors and human decision making. One of the new interdisciplinary areas currently emerging to tackle problems of this nature is the Intelligent Control Theory (Refs. 1-3) which combines conventional control theory with decision making and learning tools developed in the field of Artificial Intelligence. Following Tsypkin (Ref. 4), the necessity for applying learning arises in situations where a system must operate in conditions of uncertainty. Another active area of research is Hybrid Control Theory (Refs. 5, 6), which deals with systems that involve both continuous and discrete data structures. The discrete data may arise from sudden changes in the physical systems, from singularities in an incomplete dynamic model, from computer round off errors, or from actions controlled by higher level decision making. The discrete data often force the control system to make a choice and switch from one control law to another.

The system architecture adopted in this program has three layers: Execution, coordination, and planning. Each layer is organized further into a hierarchy of components, with the lowest level being the most precise and higher levels operating with less precise information and hence requiring an adaptive approach or learning approach. The choice between the adaptive or learning approach depends on the level of uncertainty of the problem. The adaptive approach may be sufficient for problems with less uncertainty. However, there are many different types of uncertainty, more complex uncertainty such as: when an inconsistency exists in attitude estimation which of the available sensors are more reliable than the others, which parameters in the sensor measurement model are unreliable. This type of uncertainty may have to be learned slowly through experience. This is precisely the type of situation in which the learning automaton can prove most effective. Learning in this case takes place over a long time scale relative to the normal operation of the system.

The execution level is the lowest level. It involves conventional control algorithms and interfaces to the spacecraft via sensors and actuators. The highest level consists of planners and schedulers. In a mature system, with more than one autonomous subsystems performing different functions, there may be only one planner and scheduler that manages plans and schedules tasks for all subsystems. The coordination level is the middle level, interfacing between the other two levels. This level consists of decision making tools, learning algorithms, etc. Some of these tools may be used to substitute for conventional algorithms that are too costly or too sensitive to change or uncertainty. For instance, in this paper, we apply machine learning algorithms to control the calibration process instead of using batch or sequential processes to compute sensor residuals. The learning algorithm should be independent of the physical system and of any lower level process involved. More precisely, there are many state estimator algorithms to choose from. For each calibration task scheduled, only a few of these algorithms will be chosen. These choices should have no effect on the performance of the learning algorithm.

The layer and hierarchical structure of the architecture allows us to build on an existing control system, such as ADCS, step by step beginning with ASCAL which provides attitude sensor self-calibration functionality. As development progresses, higher level adaptation is made each time a new subsystem with new functionality is added to ADCS, such as gyroscope self-calibration functionality. The new subsystem can be operated and tested independent of previously developed subsystems.

Sensor calibration problems can be viewed as a dynamical system with uncertainty in the measurement model parameters. There are several algorithms for sensor calibration (Refs. 7-10). The choice of algorithm depends on the type of sensors being considered. Typically, it is left to the attitude experts to select appropriate methods for the task. However, to demonstrate the ideas and concepts of ADCS enhanced autonomy, we will focus on only one algorithm. In a later stage of development, when the concept of self-calibration has matured, additional algorithms may be added as new subsystems in the hierarchy. Expert knowledge on algorithm selection would be coded as rules in a rule-base system in the mid-level. The rule-base will select an appropriate algorithm when a calibration task is scheduled.

An automated system such as ASCAL is useful for mission cost reduction. It automatically performs routine monitoring and trending and stores experts' knowledge of sensor and instrument calibration to be reused for future events. Moreover, ASCAL may be useful for constellation of satellites, each having similar pointing requirements. Our future extension is to apply the same architecture described here to other flight software such as orbit determination and navigation systems, tracking, and formation flying.

This paper is organized as follows. The main architecture of the system is described in Section 2. The main focus of this paper, the calibration component, is described in Section 3. The technology used in the calibration component is a heuristic learning automaton. The prioritization for the calibration process is based on the Local Dempster-Shafer theory developed in [1]. This is described in Section 4. The Coordinator and Planner level are discussed in Section 5 and 6 respectively.

2 ASCAL Architecture

Figure 1 shows the architecture of ASCAL. The execution level consists of an attitude estimator and predictor. The coordination level determines which sensor parameters need adjustment, what should their upper and lower bounds be, and which algorithms are appropriate. This level also includes the learning component in the calibration process. The planning level plans and schedules calibration tasks, making sure that computing resources are available and avoiding possible conflicts with other tasks.

It is natural to consider extended state vectors consisting of an attitude vector and erroneous sensor parameters. However, this will generally introduce additional non-linearity into the models and could make the problem intractable or too costly to run on-board. To minimize the computational cost, we apply machine learning techniques to adjust these parameters guided by past experience. Attitudes and errors are computed each time sensor parameters are adjusted. Each cycle of the computation contributes new information on the convergence of the solution. This knowledge will affect the way these parameters are adjusted.

Naively, attitude accuracy is monitored by estimating attitude using different combinations of gyros and attitude sensors, uncalibrated versus calibrated. The attitude residuals obtained from the computed attitudes are predicted using a conventional prediction algorithm. When it is discovered that the attitude residual will exceed a threshold sometime in the future, it means there is an inconsistency in the estimated attitudes. The attitude inconsistencies are then diagnosed and one or more calibration goals are created. These goals are expressed as which measurement parameters need adjustment, the range of adjustment and the most appropriate calibration algorithm. The calibration process is then planned and scheduled. In a spacecraft where one or more sensors need regular calibration, or where computing resources are limited, the predictor may be replaced by a periodic schedule managed by the planner/scheduler component. The calibration process is iterative, where the erroneous measurement parameters are adapted on the basis of system experience in such a way that the attitude inconsistencies converge to zero.

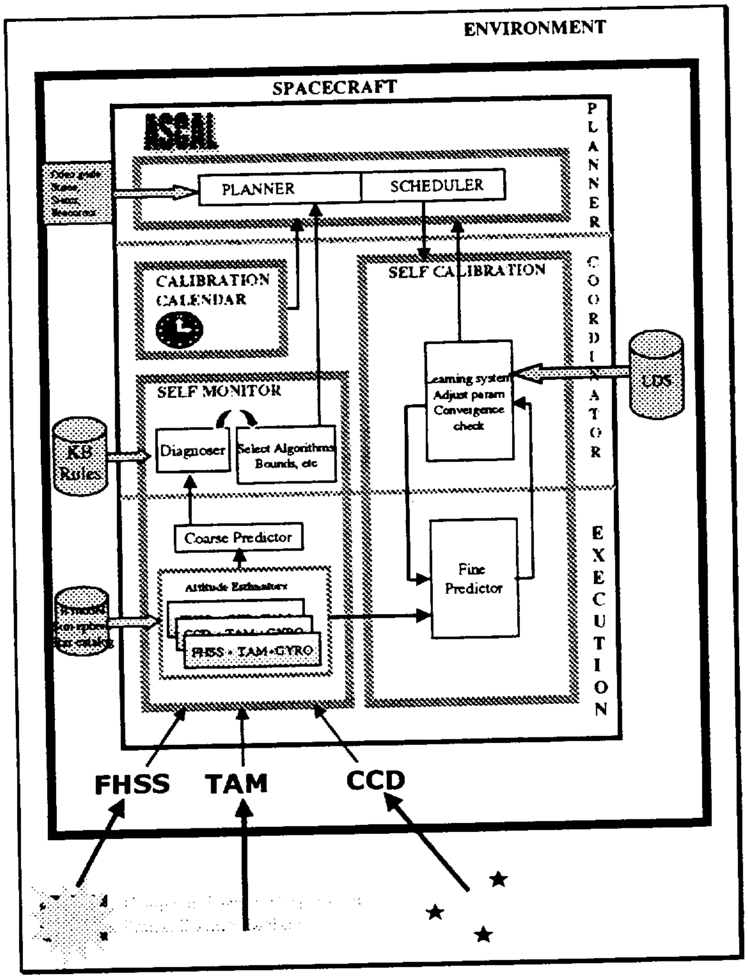


Figure 1 ASCAL architecture

3. Estimator and Predictor

When a calibration process is scheduled, the coordinator will set a goal following a guideline stored in its knowledgebase, perhaps as a set of rules. A typical goal would be to calibrate a certain set of parameters.

The calibration procedure depends on the type of sensors on-board. If there are enough redundant sensors the standard technique is to compute attitudes from a few different sets of sensors and compare the results. If the pairwise difference between these attitudes have zero mean, then there is no inconsistency, and all of the sensors are accurate (relative to each other). Generally, there are one or more sensors that are used as standard. They are the ones that have already been calibrated, or the ones with higher accuracy. We will call a set of sensors used in an attitude estimation process a *test set*. Generally, one or more of the test sets contain sensors to be calibrated, and at least one of the test set contains accurate sensors. If there are no redundant sensors, or not enough available sensors to create at least two test sets, then the calibration procedure usually involves more in depth analysis. In this paper, we assume there is at least one sensor with high accuracy, such as a Charge Coupled Device (CCD) star tracker, enabling us to calibrate other sensors against them. Such a sensor is frequently chosen as the standard frame of reference and generally does not need calibration. In this paper, we assume that there is such sensor on-board.

Before the calibration process starts, a number of test sets are identified, with at least one of the test sets containing the sensor(s) to be calibrated and the other test sets containing the standard sensor, calibrated gyros, or other high accuracy sensors. The coordinator, via its rule-base component, will also select a suitable estimator algorithm, for instance an attitude dynamic model and a measurement model for each selected sensor.

Let a denote a test set, x_a the attitude vector computed using measurements from all sensors in a. The attitude dynamics and the corresponding measurement model are

$$\begin{aligned} \dot{x}_a &= f(x_a(t)) + u_a(t) \\ z_{a_i,k} &= G_{a_i}(p_{a_i}, x_a(t_k)) + w_{a_i}(t_k), \end{aligned} \tag{1}$$

where a_i is a sensor in a, p_{a_i} is its model parameter vector. Note that, in this algorithm, each p_{a_i} is assumed constant during each estimation cycle. They are not members of state variables, however, their values will be adjusted by the learning system described in the next section.

In the following, we give a simple example of a state estimator and trend predictor to demonstrate how the learning system can be used in a calibration process. The inconsistency trend between attitude vectors associated with two different test sets a and b is the difference $T_{ab} = x_a - x_b$. The state space model for the inconsistency trend and its slope S_{ab} are

$$\begin{aligned} T_{ab}(t_{k+1}) &= T_{ab}(t_k) + S_{ab}(t_k) + v_{ab}(t_k) \\ S_{ab}(t_{k+1}) &= S_{ab}(t_k) + \omega_{ab}(t_k) \end{aligned}$$

Define a new state vector

$$X_{ab} = [T_{ab} - w_{ab} \quad S_{ab}]'$$

Then we have the following state-space model

$$\begin{aligned} X_{ab}(t_{k+1}) &= AX_{ab}(t_k) + u_{ab}(t_k) \\ T_{ab}(k) &= HX_{ab}(k) + w_{ab}(k+1) \end{aligned} \tag{2}$$

with

$$v_{ab}(t_k) = w(t_{k+1}) - w(t_k), \; u_{ab} = [w_{ab} \quad \omega_{ab}]',$$

and

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \; H = [1 \;\; 0]$$

In system (2) T_{ab} plays the role of output vector with preferred state $T_{ab,0} = 0$. All of the process and measurement noises are assumed to be white Gaussian with zero mean. The systems (1) and (2) become a two-stage problem. Given a set of sensor parameters, attitudes are computed either by a batch least square or a sequential method. System (2) predicts the inconsistency trend. Here we write it as a single step predictor, but a multiple step predictor can also be done.

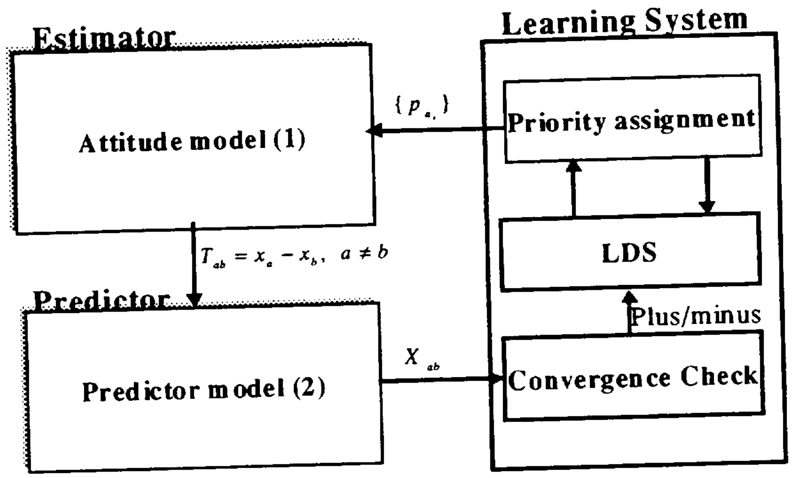


Figure 2. ASCAL Learning System

4 Learning Systems

The heart of a learning system is the learning algorithm which is the mechanism used to adapt the probability distribution. Based on the environment response and the action selected by the system at time *t*, it generates *p(t+1)* from *p(t)*. There are two levels of learning in ASCAL. When a calibration goal is set, the coordinator must determine the sensors, algorithms, and parameter ranges needed to initialize the calibration process. This selection is based on the past experiences. In particular, the parameter ranges are chosen in such a way that the region is void of any singularity and at least one solution exists. This knowledge can be given *a priori* by attitude experts, and maintained by a learning algorithm. The second level of learning is in the calibration process, where attitude residuals are computed, convergence tested, and parameters adjusted sequentially. We assume that an appropriate metric is defined on the state space. The selection of parameter adjustment is a learning process based on the rate of convergence (or divergence) of the attitude residuals during the previous two (or more) cycles. Assume there are *n* sensor parameters to be adjusted, and each parameter can be increased or decreased by a fixed quantity. This corresponds to $H = \sum_{i=0}^{n} 2^i \left(\frac{n}{i} \right)$ possible actions, where each action is a set of parameters, each associated with a + or − sign to denote if it is increased or decreased. For instance, an action corresponding to an increase in *a* and decrease in *b* is represented by the signed set $\{a_+, b_-\}$. These *H* actions are prioritized by a probability or belief vector given by the Local Dempster-Shafer (LDS) (Ref s 11, 12).

To get a feel for the learning algorithm based on LDS, we will now describe a simpler algorithm based on the Dempster-Shafer (DS) theory (Refs. 13, 14), modified to suit our calibration problem. For a more in depth discussion of the LDS theory see Ref. 12. DS theory is defined on a set of *n* elements. A mass function on the action set *H* is a probability function that assigns a degree of belief to each action. More precisely, the mass function satisfies the following conditions

$$\sum_{A \supseteq H} m(A) = 1, \text{for } A \neq \emptyset \quad \text{and} \qquad m(\emptyset) = 0$$

Two mass functions m_1 and m_2 on *H* can be combined into a single mass function $m_1 \otimes m_2$ by the Dempster composition rule:

$$m_1 \otimes m_2(A) \;=\; \sum_{B\cup C=A} m_1(B)m_2(C) \Big/ (1-\sum_{B\cup C=\varnothing} m_1(B)m_2(C)), \qquad \text{for } A\neq\varnothing$$

$$m_1 \otimes m_2(\varnothing)=0.$$

The belief function associated to the mass function m is defined to be the cumulative probability distribution on H:

$$b: H \rightarrow [0,1];\; b(A)=\sum_{B\supseteq A} m(B)$$

where the union between two signed sets is defined as the union of all signed elements, followed by removing every subset of the form $\{a_+, a_-\}$ for some parameter a. The belief function is used to prioritize the actions for the learning algorithm. If an action is chosen and the resulting attitude residuals decrease with a faster rate or increase with a slower rate, then the system reprioritizes by applying the positive learning algorithm described in Ref 12. This will strengthen the previous prioritization. Conversely, if the previously chosen action is performed in the opposite manner, then the system reprioritizes by applying negative learning algorithm, which will lessen the degree of belief on the failed action.

In general, a learning system may have a hierarchical structure. In this case, the selection of the action set should also have a hierarchical structure. To support this structure, a hierarchical flavor of DS theory can be defined in a natural way. The action selection is performed in a sequence of steps. First, a highest level in the hierarchy of the action set is selected, followed by a lower action. This procedure is followed until the last level of the action set H. This hierarchical structure will clearly reduce the size of the search space, and hence enhance the performance of the automata.

The learning process discussed above is the simplest application of the (modified) DS theory to learning automata. In practice this algorithm can be enhanced in several different ways to increase the performance and robustness of the learning system. Our possible future research topics in this areas are: Localization of the action space (H) by applying LDS theory instead of DS theory. This will reduce the complexity of the search and increase the performance of ASCAL. Instead of keeping the step size of parameter modification constant, we may consider it as a function of the rate of convergence computed from the previous cycles. The function that works will guarantee the convergence of the solution. The use of hierarchical or multilevel learning systems accelerates the learning process (more so for the initial rate of learning) and simplifies the structure of the learning system. The learning system discussed above is an active research area with many applications in intelligent and hybrid control problems.

5 Coordinator

In some sense, the coordinator is a process manager whose responsibility is to monitor the physical subsystem it is responsible for, i.e. the ADCS, and predict if any problem, i.e. an attitude inconsistency, will occur. If a problem is predicted, the coordinator will identify the source of the problem and create goals to solve it.

The responsibility of the coordinator consists of two parts: monitoring/diagnosis and pre-calibration. The monitoring/diagnosis components monitors the state of health of the ADCS by periodically computing and trending relative attitude residuals using multiple test sets. When an attitude inconsistency is predicted, the diagnosis component determines which sensor parameters are likely to be unreliable based on attitude data that displays the trends. The result of the diagnosis is the degree of unreliability, a probabilistic quantity, assigned to each sensor parameter involved in the trending process. Underlying the diagnosis process is the uncertainty handler based on the LDS theory, (Ref 11). When this is done, the coordinator creates goal to calibrate the problematic parameters, and submits the goal to the planner.

When a calibration process is scheduled the pre-calibration tasks begin. First, based on the degree of unreliability, a collection of test sets is formed, and the bounds for the sensor parameters are computed. Based on the sensors involved, attitude dynamics and measurement models are selected, and a state-space system is defined for each test set. Finally, the coordinator also determines any *a priori* knowledge the calibration process may need, including the initial probability distribution for the learning system to use as priority assignment in the learning process. It is convenient to use the degree of unreliability as the initial probability distribution. However, other expert knowledge the system may have can be combined with the degree of unreliability using a modified Dempster combination rule. The coordinator performs these tasks using decision making capability such as a rule-base.

6 Planner & Scheduler

This component may be responsible for several autonomous systems. For ASCAL, the planner/scheduler is responsible for scheduling sensor calibration. It should be aware of available sensors, i.e. those with target in field of view, and related resources. This means the spacecraft must be sufficiently equipped with a star catalog, and Sun, Earth, and Moon ephemerides. The system must also be able to perform some maneuver planning needed for gyroscope calibration, or to sample selected targets throughout the field of view.

In this version of ASCAL, the calibration process uses live data from attitude sensors on-board, to avoid dealing with attitude history data management which is a formidable problem of its own. However, as a trade off, the planner will have to be smart enough to avoid conflicts among spacecraft activities, to manage resources such as available sensors and computer time. A simple solution is to find a quiet window of time when there is no important activity on-board and devote all attention to the calibration process.

7 Conclusion and Implementation Status

This study is the first phase of our program to extend the degree of autonomy of on-board flight software. The consequences of failure are catastrophic for an attitude control system. If the attitude control system fails for even a brief period, the spacecraft may tumble, pointing the solar arrays away from the sun, antennas away from the earth, and sensitive instrumentation in a potentially damaging direction. Such a control system failure may or may not be recoverable. Nonetheless, virtually all spacecraft have fully autonomous, onboard attitude control. Failure to properly calibrate the sensor parameters would lead to inaccuracies in attitude estimation, and would in turn lead to attitude control system failure. Sensor calibration is traditionally done from the ground, because the standard procedures and algorithms are storage and computationally intensive. In this paper, we propose a non traditional approach, using learning automata and heuristic priority assignment to adjust sensor parameters until all inconsistencies converge to within an acceptable limit. Human intervention is called for if this process does not converge, and if the diagnoser cannot resolve the problem. In this case, the lessons learned should be added into the knowledgebase for future use. It is important to design the learning algorithms so that they are independent of the sensors being calibrated or of changes in the environment. This is key for autonomous attitude sensor calibration in future missions.

The next natural step towards higher level on-board automation is to add data management capability to ASCAL. Calibration process can be performed using historical data without disturbing other activities, except computer resources. To archive measurement data for the calibration process ahead of time would require, the coordinator can be augmented with a data processing component. It is responsible for data pre or post processing, data smoothing, and/or shifting. Generally, measurement data are sensitive to some spacecraft's activities such as maneuvering. The planner/scheduler must be aware of these activities. With this knowledge obtained from the planner/scheduler, the data processor may avoid the disturbed data, so that relatively clean data for the past, say 24 hours, may be stored and ready to be used when a calibration process is scheduled. This problem suggests that high level autonomy is necessary for autonomy system development such as ASCAL.

Other possible future development is autonomous orbit determination and control, orbit keeping, maneuvering, and formation flying. Machine learning approach described in this paper is a generic tool that is likely to be useful in these applications.

References

1. P. J. Antsaklis, and K. M. Passino, eds., *An Introduction to Intelligence and Autonomous Control*, Norwell, MA: Kluver Academic Publishers, 1993.
2. P. J. Antsaklis, K. M. Passino, and S. J. Wang, *An Introduction to Autonomous Control Systems*, Special Issue on Intelligence Control, IEEE Control Systems Magazine, 11, 4, 5-13 1991.
3. P. J. Antsaklis, K. M. Passino, and S. J. Wang, *Towards Intelligent Autonomous Control System: Architecture and Fundamental Issues*, Journal of Intelligent and Robotic Systems, 1, 4, 315-342, 1989.
4. Y. T. Tsypkin, *Foundation of the Theory of Learning Systems*, Academic Press 1973.
5. P. J. Antsaklis, M. D> Lemmon, and J. A. Stiver, *Hybrid System Modeling and Event Identification*, Proc of the 29th Allerton Conf. On Communication, Control, and Computing. 64-73, University of Illinois at Urbana-Champaign, Sep. 1993.
6. A. Gollu, and P. Varaiya, *Hybrid Dynamical Systems*, Proc. Of the 28th Conf on Decision and Control, 2708-2712, Tampa, FL, Dec. 1989.
7. M. D. Shuster, D. M. Chitre, and D. P. Niebur, *Inflight Estimation of Spacecraft Attitude Sensor Accuracies and Alignments*, J. of Guidance and Control, 5, 4, 1982.
8. M. D. Schuster, *In-Flight Estimation of Spacecraft Sensor Alignment*, Advances in the Astronautical Sciences, 72, 1990.
9. G. J. Bierman and M. D. Schuster, *Spacecraft Alignment Estimation*, 27th IEEE Conference on Decision and Control, Austin, TX, Dec. 7-9 1988.
10. J. E. Keat, *Gyro Calibration Analysis for the High Energy Astronomy Observatory-A (HEAO-A)*, Computer Sciences Corporation, CSC/TM-77/6082, June 1977.
11. C. Sary, C. Peterson, J. Rowe, K. Mueller W. Truszkowski and N. Ziyad, *Trend Analysis for Spacecraft Systems Using Multimodal Reasoning*, 152-158, AAAI Spring Symposium, Stanford, Palo Alto, CA, Mar. 1998.
12. C. Peterson, K. Mueller, *Local Dempster Shafer Theory*, submitted to the J. of Experimental and Theoretical Artificial Intell
13. G. A. Shafer, *Mathematical Theory of Evidence*, Princeton U. Press, 1976.
14. A. P. Dempster, *Upper and Lower Probabilities Induced by Multivalued Mappings*, Annals of Math. Stat. 38, pp. 325-329, 1967.

ANGULAR-RATE ESTIMATION USING STAR TRACKER MEASUREMENTS

R. Azor[*], Itzhack Y. Bar-Itzhack[+], Julie K. Deutschmann[#],
and Richard R. Harman[#]

ABSTRACT

This paper presents algorithms for estimating the angular-rate vector of satellites using quaternion measurements. Two approaches are compared, one that uses differentiated quaternion measurements to yield coarse rate measurements which are then fed into two different estimators. In the other approach the raw quaternion measurements themselves are fed directly into the two estimators.

The two estimators rely on the ability to decompose the non-linear rate dependent part of the rotational dynamics equation of a rigid body into a product of an angular-rate dependent matrix and the angular-rate vector itself. This decomposition, which is not unique, enables the treatment of the nonlinear spacecraft dynamics model as a linear one and, consequently, the application of a Pseudo-Linear Kalman Filter (PSELIKA). It also enables the application of a special Kalman filter which is based on the use of the solution of the State Dependent Algebraic Riccati Equation (SDARE) in order to compute the Kalman gain matrix and thus eliminates the need to propagate and update the filter covariance matrix. The replacement of the elaborate rotational dynamics by a simple first order Markov model is also examined.

In this paper a special consideration is given to the problem of delayed quaternion measurements. Two solutions to this problem are suggested and tested.

Real Rossi X-Ray Timing Explorer (RXTE) data is used to test these algorithms, and results of these tests are presented.

I. INTRODUCTION

In most spacecraft (SC) there is a need to know the SC angular-rate. Precise angular-rate is required for attitude determination, and a coarse rate is needed for attitude control damping. Classically, angular-rate information is obtained from gyro measurement. Over the past few years, there has been a tendency to build smaller, lighter and cheaper SC. One means of building less costly SC is to use cheaper sensors or even fewer sensors. One very expensive sensor most spacecraft will use is the gyro. An alternative means of estimating a high fidelity rate would provide a significant cost savings as well as provide for angular rate estimation during high rate anomalies which might be beyond the range of the purchased gyros.

There are several ways to obtain the angular-rate in a gyro-less SC. When the attitude is known, one can differentiate the attitude in whatever parameters it is given and use the kinematics equation that connects the derivative of the attitude with the satellite angular-rate in order to compute the latter[1]. Since SC usually utilize vector measurements for attitude determination, the differentiation of the attitude introduces a considerable noise component in the computed angular-rate vector. To overcome this noise, the computed rate components can be filtered by a passive low pass filter. This, however, introduces a delay in the computed rate[1]. When using an active filter, like a Kalman filter (KF), the delay can be eliminated[2,3].

Another approach may also be adopted for the problem of angular-rate computation where the vector measurements themselves are differentiated.

[*] Senior Control Engineer, Systems & Space Technology, Electronics Div., MBT, Systems Eng. Dept., Israel Aircraft Industries, Inc. Yehud 56000, Israel.

[+] Sophie and William Shamban Prof. of Aerospace Eng., Faculty of Aerospace Eng., Technion-I.I.T., Member Technion Asher Space Research Inst.. AIAA Fellow, IEEE Fellow. Haifa 32000, Israel. Tel: +972-4-829-3196. Fax: +972-4-823-1848. Email address: ibaritz@tx.technion.ac.il

[#] Aerospace Engineer, Flight Dynamics Analysis Branch, Code 572, NASA Goddard Space Flight Center Greenbelt, MD 20771.

This approach was used by Natanson[4] for estimating attitude from magnetometer measurements, and by Challa, Natanson, Deutschmann and Galal[5] to obtain attitude as well as rate. Similarly, Challa, Kotaru and Natanson[6] used derivatives of the earth magnetic field vector to obtain attitude and rate.

All these methods use the derivative of either the attitude parameters or of the measured directions which normally determine the attitude parameters. Another approach is that of using the attitude parameters, or the measured directions themselves, as measurements in some kind of a KF. In this case the kinematics equation that connects the attitude parameters, or the directions, with their derivatives are included in the dynamics equation used by the filter thereby, as will be shown in the ensuing, *the need for differentiation is eliminated*[7,8].

New sensor packages have been introduced lately that yield the SC attitude in terms of the attitude quaternion[9]. Therefore it is possible to use the quaternion supplied by such sensors as measurements and, as mentioned before, eliminate the need for differentiation. In this paper we investigate this possibility.

As mentioned, in the ensuing we will apply two special KFs which make use of the SC angular dynamics model; therefore, by way of introduction, in the next section we present the development of the SC dynamics model, and in Section III we present the two filters. For comparison purposes, in Section IV we treat the approach where the angular-rate is still extracted from derivative but here we pass the resultant noisy quaternion through the two active rather than through a passive filter as was done in Ref. 2. The other approach, where the raw quaternion measurements themselves are fed into the filter, requires the addition of the quaternion to the state vector which is comprised of the angular-rate vector. This is treated in Section V. In Section VI we consider the case where the filter dynamics is drastically simplified by reducing the dynamics equation of the SC to a first order Markov process. The issue of quaternion normalization is presented in Section VII, and in Section VIII we solve the problem of measurement delay. The last section of this work is the Conclusion section.

II. FILTER DYNAMICS MODEL

The main dynamics model is that which describes the propagation of the SC angular velocity, $\underline{\omega}$. The angular dynamics of a constant mass SC is given in the following equation[10]

$$\dot{\underline{\omega}}I + \dot{\mathbf{h}} + \underline{\omega} \times (I\underline{\omega} + \mathbf{h}) = \mathbf{T} \qquad (1)$$

where $\underline{\omega}^T = [\omega_x, \omega_y, \omega_z]$, I is the SC inertia tensor, **h** is the momentum of the momentum wheels, and **T** is the external torque operating on the SC. The components ω_x, ω_y and ω_z are the three components of the sought angular-rate vector, $\underline{\omega}$, of the SC body with respect to inertial space when resolved in the body coordinates. Eq. (1) can be written as

$$\dot{\underline{\omega}} = I^{-1}[(I\underline{\omega} + \mathbf{h})\times]\underline{\omega} + I^{-1}(\mathbf{T} - \dot{\mathbf{h}}) \qquad (2)$$

where $[(I\underline{\omega} + \mathbf{h})\times]$ is the cross product matrix of the vector $(I\underline{\omega} + \mathbf{h})$. Define

$$F(\underline{\omega}) = I^{-1}[(I\underline{\omega} + \mathbf{h})\times] \qquad (3)$$

and

$$\mathbf{u}(t) = I^{-1}(\mathbf{T} - \dot{\mathbf{h}}) \qquad (4)$$

then Eq. (2) can be written in the form

$$\dot{\underline{\omega}} = F(\underline{\omega})\underline{\omega} + \mathbf{u}(t) \qquad (5)$$

As was shown in Ref. 2, there are 8 primary models, and infinite linear combinations of them, which express Eq. (1) in the form of Eq. (5).

Eq. (5) describes the SC correct dynamics; however, we usually do not know the exact values of I, T , **h** and its derivative, therefore we do not know the exact relationship between $\dot{\underline{\omega}}$ and these elements. We express our lack of knowledge by adding a stochastic process to the dynamics equation of Eq. (1). We assume that this stochastic process, w(t), is a zero mean white noise process. The resulting model which is used by the estimator is

$$\dot{\underline{\omega}} = F(\omega)\underline{\omega} + \mathbf{u}(t) + \mathbf{w}'(t) \qquad (6)$$

If we denote $\underline{\omega}$ by **x**, then Eq. (6) can be written as

$$\dot{\mathbf{x}} = F(\mathbf{x})\mathbf{x} + \mathbf{u}(t) + \mathbf{w}'(t) \qquad (7)$$

where obviously

$$F(\mathbf{x}) = I^{-1}[(\mathbf{Ix} + \mathbf{h})\times] \qquad (8)$$

For the time being we assume that we measure the angular-rate; that is, **x**, therefore the measurement equation is

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k \qquad (9)$$

where

$$H = I_3 \qquad (10)$$

v_k is a zero mean white measurement noise, and I_3 is the third dimensional identity matrix.

III. ANGULAR-RATE ESTIMATION

As mentioned in the introduction section, we use two filtering algorithms to estimate the angular-rate. These algorithms are described next.

The dynamics equation presented in Eq. (7) is a nonlinear differential equation due to the term $F(\mathbf{x})\mathbf{x}$. A standard filter for this case is the Extended Kalman Filter (EKF). One can also apply the Extended Interlaced Kalman filter[3] where three linear KFs are run in parallel. Other possibilities which are applicable to the form of non-linearity presented in Eq. (7) are the Pseudo-Linear Kalman (PSELIKA) filter and the State Dependent Algebraic Riccati Equation (SDARE) filter which were used successfully in Ref. 2. In view of their performance, the latter two filters are used in this work too.

III.1 The Pseudo-Linear Kalman (PSELIKA) Filter

The PSELIKA filter algorithm disregards the non-linearity and treats the dynamics system as if it were just a time varying system, consequently, the ordinary KF algorithm is applied. First, the continuous differential equation (7) expressing the SC dynamics is discretized and then the KF algorithm is applied as follows. First evaluate:

$$W'_k = E\{\mathbf{w}'(t_k)\mathbf{w}'(t_k)^T\} \qquad (11)$$

$$R_k = E\{\mathbf{v}_k\mathbf{v}_k^T\} \qquad (12)$$

and choose an approximate value for the initial estimate of the rate vector. In the absence of such initial estimate, choose $\hat{\mathbf{x}}_0 = 0$. Next, determine P_0, the initial covariance matrix of the estimation error according to the confidence in the choice of $\hat{\mathbf{x}}_0$. The recurrence algorithm is then as follows.

- ***time propagation***:

Let A_k be the discrete dynamics matrix obtained when $F(\mathbf{x})$ of Eq. (8) is discretized, and let $\mathbf{u}_k$ be the discrete deterministic input signal, then propagate the state estimate according to:

$$\hat{\mathbf{x}}_{k+1/k} = A_k\hat{\mathbf{x}}_{k/k} + \mathbf{u}_k \qquad (13.a)$$

and the covariance matrix according to:

$$P_{k+1/k} = A_kP_{k/k}A_k^T + W'_k \qquad (13.b)$$

- ***measurement update***:

Compute the Kalman Gain as follows:

$$K_{k+1} = P_{k+1/k}H^T[HP_{k+1/k}H^T + R_{k+1}]^{-1} \qquad (13.c)$$

Update the estimate according to:

$$\hat{\mathbf{x}}_{k+1/k+1} = \hat{\mathbf{x}}_{k+1/k} + K_{k+1}[\mathbf{z}_{k+1} - H\hat{\mathbf{x}}_{k+1/k}] \qquad (13.d)$$

and update the covariance matrix using:

$$P_{k+1/k+1} = [I - K_{k+1}H]P_{k+1/k}[I_3 - K_{k+1}H]^T + K_{k+1}R_{k+1}K_{k+1}^T \qquad (13e)$$

III.2 The State Dependent Algebraic Riccati Equation (SDARE)

The continuous-discrete-time SDARE filter which was used in Ref. 2 was based on the work of Cloutier, D'Souza and Mracek[11,12], Pappano and Friedland[13], and Mracek, Cloutier and D'Souza[14]. That continuous-discrete-time filter for the continuous-time dynamics and the discrete-time measurement is as follows (see Ref. 2).

As with the PSELIKA filter, choose an approximate value for the initial estimate of the rate vector. In the absence of such initial estimate, choose again $\hat{\mathbf{x}}_0 = 0$.

- ***time propagation***:

Propagate the state estimate according to:

$$\hat{\mathbf{x}}_{k+1/k} = A_k\hat{\mathbf{x}}_{k/k} + \mathbf{u}_k \qquad (14)$$

- ***measurement update***:

At the measurement updating time, t_{k+1}, solve the following algebraic Riccati equation for P_{k+1}:

$$A(\hat{\mathbf{x}}_{k+1/k})P_{k+1} + P_{k+1}A^T(\hat{\mathbf{x}}_{k+1/k}) - P_{k+1}H^TR_{k+1}^{-1}HP_{k+1} + W'_{k+1} = 0 \qquad (15.a)$$

and compute the gain matrix:

$$K_{k+1} = P_{k+1}H^T R_{k+1}^{-1} \qquad (15.b)$$

Finally compute the updated state estimate:

$$\hat{\mathbf{x}}_{k+1/k+1} = \hat{\mathbf{x}}_{k+1/k} + K_{k+1}[\mathbf{z}_{k+1} - H\hat{\mathbf{x}}_{k+1/k}] \qquad (15.c)$$

IV. THE FILTERED QUATERNION-RATE APPROACH

As mentioned before, it is possible to derive $\underline{\omega}_r$, a crude estimate of $\underline{\omega}$ using the quaternion first time-derivative[1,2]; however, the resultant estimate is noisy. If $\underline{\omega}_r$ is passed through a passive low-pass filter the noise may be filtered out at the expense of a delay[1]. Here we investigate the quality of the filtered rates when the two active filters described before are used to filter $\underline{\omega}_r$. First we show how $\underline{\omega}_r$ is derived from $\dot{\mathbf{q}}$, the differentiated quaternion. As is well known [see e.g. Ref. 10], the quaternion dynamics equation is

$$\dot{\mathbf{q}} = \tfrac{1}{2}\Omega\mathbf{q} \qquad (16)$$

where

$$\Omega = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \qquad (17)$$

It is also known [see e.g. Ref. 1] that Eq. (16) can be written as

$$\dot{\mathbf{q}} = \tfrac{1}{2}Q\omega \qquad (18)$$

where

$$Q = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \qquad (19)$$

Define the pseudo inverse

$$Q^{\#} = (Q^TQ)^{-1}Q^T \qquad (20.a)$$

where T denotes the transpose. Note that

$$Q^TQ = I_3 \qquad (20.b)$$

where I_3 is the fourth dimensional identity matrix. From Eqs. (18) and (20.a) it is easily seen that a rough estimate of the rate vector can be computed as follows

$$\underline{\omega}_r = 2Q^{\#}\dot{\mathbf{q}} \qquad (21.a)$$

which in view of Eqs. (20) can be written simply as

$$\underline{\omega}_r = 2Q^T\dot{\mathbf{q}} \qquad (21.b)$$

The dynamics equation for the estimator was introduced in Section II (see Eq. 7); thus, in view of Eq. (21), like Eq. (10), the measurement equation which corresponds to that dynamics model is

$$\underline{\omega}_r = H_\omega\underline{\omega} + v_\omega \qquad (22.a)$$

where

$$H_\omega = I_3 \qquad (22.b)$$

and $\mathbf{v}_\omega$ is a zero mean white noise.

The Pseudo-Linear Kalman Filter (PSELIKA) and the State Dependent Algebraic Riccati Equation (SDARE) filter were used to obtain the angular-rate from quaternion observations using the Quaternion-Rate approach. The data which was used to test this approach was real measurements downloaded from the RXTE satellite, which was launched on Dec. 30, 1995. We chose a segment of data starting January 4, 1996 at 21 hours, 30 minutes, and 1.148 sec. The quaternion which was used was based on the SC attitude as determined by its star trackers. Fig. 1 presents $\underline{\omega}$, the nominal angular-rate, Fig. 2 presents the error between $\underline{\omega}_r$, the raw angular-rate, and $\underline{\omega}$, the nominal rate In order to quantify the error, a single figure of merit (FM) is computed. First the average square error of each component is computed as follows $\overline{e_i^2} = \frac{1}{T-t_0}\int_{t_0}^{T} e_i^2 dt$ i=x, y, z. This computation yields $\overline{e_x^2}, \overline{e_y^2}$ and $\overline{e_z^2}$. Then the FM is computed as $FM = \sqrt{\overline{e_x^2} + \overline{e_y^2} + \overline{e_z^2}}$. In order to exclude the transients we set $t_0 = 100\,\text{sec}$. It was found that $FM(2) = 7.3998 \cdot 10^{-3}\,\text{deg/sec}$ where FM(2) is the FM of Fig. 2. Fig. 3 presents the estimation error when the PSELIKA filter was applied to $\underline{\omega}_r$. It was found that $FM(3) = 1.5311 \cdot 10^{-3}\,\text{deg/sec}$. Finally, Fig. 4 shows the same when the SDARE filter was used and it was found that $FM(4) = 1.4550 \cdot 10^{-3}\,\text{deg/sec}$. As indicated by FM(2), the computed angular-rate, $\underline{\omega}_r$, particularly its x component, was rather noisy. When either the PSELIKA or the SDARE filter were applied to $\underline{\omega}_r$, other than a few spikes, the resulting $\hat{\underline{\omega}}$ was smoother. In this example there was no real difference between the performance of the two filters (see FM(3) and FM(4)). As expected, the computation of $\underline{\omega}_r$ using Eq. (21) produced a noisy estimate due to the differentiation of the measured quaternion which was corrupted by measurement noise, and the application of the PSELIKA filter to this $\underline{\omega}_r$ filtered out most of the noise. When the SDARE rather than the PSELIKA filter was applied to $\underline{\omega}_r$, the filtered estimate of the angular-rate was visually identical. In other words, the effect of the application of the SDARE filter was practically identical to that of the PSELIKA filter.

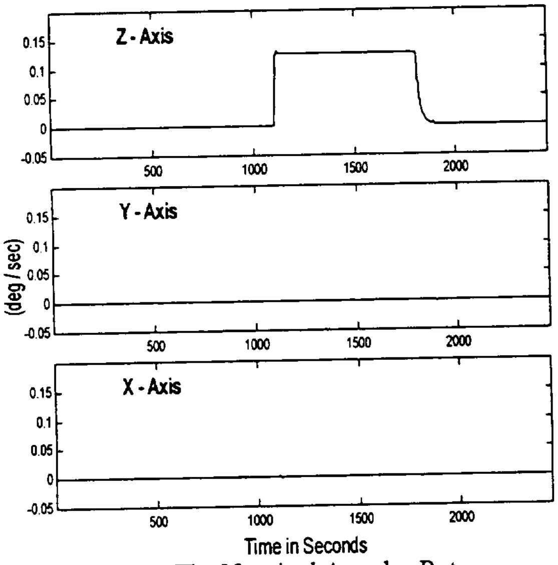


Fig. 1: The Nominal Angular-Rate.

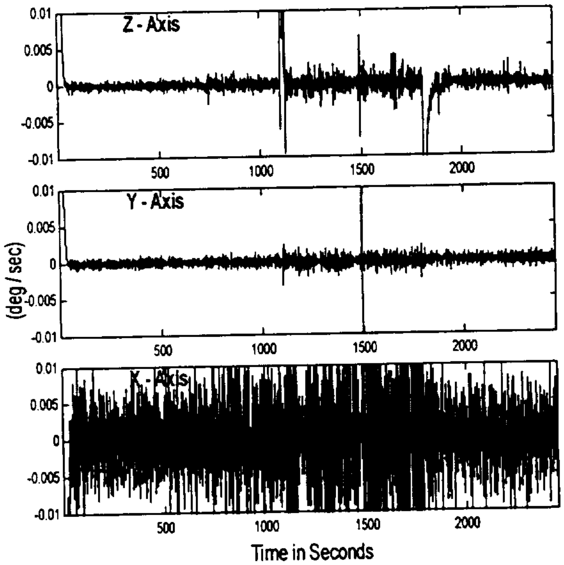


Fig. 2: The Error Between the Raw Angular-Rate, $\underline{\omega}_r$, and the Nominal Angular-Rate .

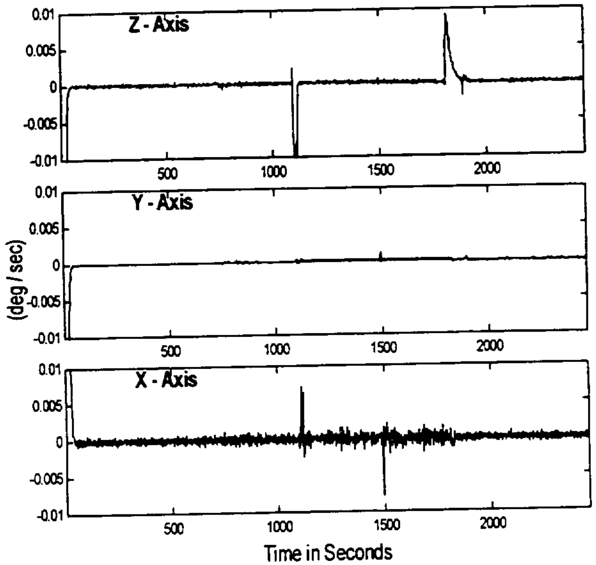


Fig. 3: The Error in the Estimated Angular-Rate, $\hat{\underline{\omega}}$, After the PSELIKA Filter Applied to $\underline{\omega}_r$.

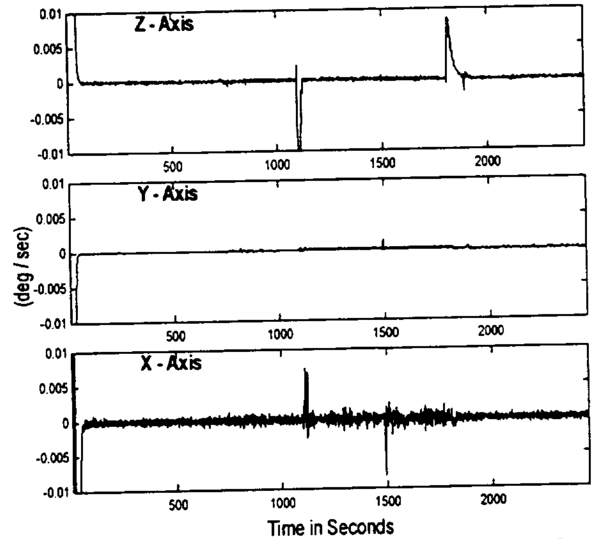


Fig. 4: The Error in the Estimated Angular-Rate, $\hat{\underline{\omega}}$, After the SDARE Filter was Applied to $\underline{\omega}_r$.

V. THE QUATERNION AUGMENTATION APPROACH

Although we also tested the Quaternion-Rate approach described in the preceding section, in this work we are mainly interested in estimating $\underline{\omega}$ *using the measured quaternion itself* rather than its derivative. However, the quaternion is not a part of the state vector of the system (see Eqs. 6, 7). One solution to this problem was examined in the preceding section. Another solution is the augmentation of the quaternion with the angular-rate state of Eqs. (6, 7). For this we can use the quaternion dynamics equation given in Eq. (16) and obtain the following model which augments Eqs. (6) and (16)

$$\dot{y} = G'(y)y + e(t) + g(t) \qquad (23)$$

where

$$y = \left[\begin{array}{c}\underline{\omega}\\ \hline q\end{array}\right] \qquad (24.a)$$

$$G'(y) = \begin{bmatrix}F(\underline{\omega}) & 0\\ 0 & \frac{1}{2}\Omega\end{bmatrix} \qquad (24.b)$$

$$e(t) = \begin{bmatrix}u(t)\\ 0\end{bmatrix} \qquad (24.c)$$

$$g(t) = \begin{bmatrix}w'(t)\\ 0\end{bmatrix} \qquad (24.d)$$

The measurements of the quaternion are taken at discrete time points; therefore the measurement model is a discrete one. The discrete measurement model that corresponds to the dynamics model of Eq. (23) is

$$q_{m,k} = C\left[\begin{array}{c}\underline{\omega}\\ \hline q\end{array}\right]_k + v_k \qquad (25)$$

where $q_{m,k}$ is the measurement at time t_k,

$$C = \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} \qquad (26)$$

and v_k is the measurement noise at that time.

An inspection of the matrices $G'(y)$ of Eq. (24.b) and C of the last equation reveals that even when $\underline{\omega}$ is constant this pair is deterministically unobservable. This problem can be overcome though using the fact that Eq. (16) can be written as Eq. (18) which can also be written as

$$\dot{q} = [\tfrac{1}{2}Q \,|\, 0]\left[\begin{array}{c}\underline{\omega}\\ \hline q\end{array}\right] \qquad (27)$$

therefore Eq. (23) can be transformed into

$$\dot{y} = G(y)y + e(t) + g(t) \qquad (28.a)$$

where

$$G(y) = \begin{bmatrix}F(\underline{\omega}) & 0\\ \frac{1}{2}Q & 0\end{bmatrix} \qquad (28.b)$$

We note that the measurement equation (see Eq. 25) is unchanged although the dynamics matrix of the system changes from $G'(y)$ to $G(y)$. Unlike the pair $G'(y)$ and C, the pair $G(y)$ and C is not necessarily deterministically unobservable. In fact, the results which are presented in Fig. 5 show that the pair is observable even when $\underline{\omega}$ is time varying. Moreover, in the computation of Ω which is needed in Eq. (24.b) we use our best estimate of $\underline{\omega}$. At least initially, this estimate may be way off yielding a wrong Ω and, consequently, a wrong $G'(y)$. On the other hand, in the computation of $G(y)$, given in Eq. (27.b), we use Q rather than Ω, and since Q is based on the computed **q** which is fairly accurate, we obtain a pretty accurate $G(y)$. In other words, not only is the pair $\{G(y), C\}$ observable, the use of $G(y)$ yields a more accurate filter model than does $G'(y)$. The FM of Fig. 5 was found to be $FM(5) = 6.1839 \cdot 10^{-4}$ deg/sec. When comparing Fig. 5 to Figs. 3 and 4 it is realized that the addition of **q** to the state vector yields a better filter. It is

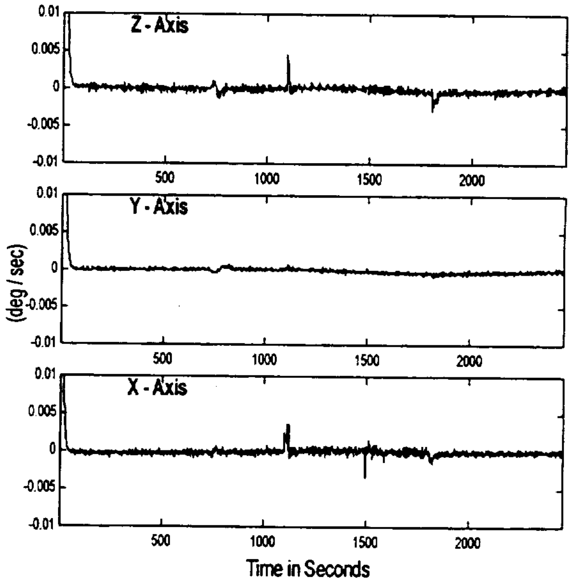


Fig. 5: The Estimated Angular-Rate, $\hat{\underline{\omega}}$, After Applying the PSELIKA Filter to the Augmented Model.

noted that the level of the spikes present in Figs. 3 and 4 was reduced when this filter was used.

While $\mathbf{v}_k$, the measurement noise vector, can be assumed to be statistically independent over time, its components are correlated with one another; moreover, it cannot be assumed that $\mathbf{v}_k$ has a constantly zero mean, consequently we model the measurement noise as

$$\mathbf{v}_k = \mathbf{v}_{1,k} + \mathbf{v}_{2,k} \qquad (29)$$

where between the measurement points , k-1, k, k+1, the noise component, v_1, changes according to

$$\dot{\mathrm{v}}_1 = -\mathrm{N}\mathrm{v}_1 + \underline{\mu}_1 \qquad (30)$$

It is further assumed that $\mathbf{v}_{2,k}$ is a zero mean white noise process whose covariance matrix contains, in general, non-zero off diagonal elements. As usual, the covariance matrix of the white noise vector, $\underline{\mu}_1$, which drives $\mathbf{v}_{1,k+1}$, is selected[15] to fit the covariance matrix of $\mathbf{v}_{1,k+1}$. That matrix too may have non-zero off diagonal elements in order to generate the correct covariance between the components of $\mathbf{v}_{1,k+1}$.

Since the measurement noise has a non-white component, one needs to augment the non-white state with the existing state vector to form a new augmented state. The resultant model is then as follows

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{f} + \mathbf{w} \qquad (31)$$

where

$$\mathbf{x} = \begin{bmatrix} \underline{\omega} \\ \mathbf{q} \\ \mathbf{v}_1 \end{bmatrix} \quad (32.a) \qquad \mathrm{F} = \begin{bmatrix} \mathrm{F}(\underline{\omega}) & 0 & 0 \\ \frac{1}{2}\mathrm{Q} & 0 & 0 \\ 0 & 0 & -\mathrm{N} \end{bmatrix} \quad (32.b)$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (32.c) \qquad \mathbf{w} = \begin{bmatrix} \mathbf{w}' \\ \mathbf{0} \\ \boldsymbol{\mu}_1 \end{bmatrix} \quad (32.d)$$

Since

$$\mathbf{q}_{m,k} = \mathbf{q}_k + \mathbf{v}_{1,k} + \mathbf{v}_{2,k} \qquad (33.a)$$

then the corresponding discrete measurement equation is

$$\mathbf{z}_{k+1} = \mathrm{H}\mathbf{x}_{k+1} + \mathbf{v}_{2,k+1} \qquad (33.b)$$

where

$$\mathrm{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (33.c)$$

VI. A SIMPLIFIED FILTER MODEL

The dynamics models which were used in the preceding section can sometimes be drastically simplified by exchanging the SC non-linear dynamics model with a simple first order Markov model. This approach, which is common practice in target tracking, was applied recently to attitude determination[7] and is considered here. The simplified filter dynamics equation takes the form

$$\mathrm{F}_s = \begin{bmatrix} -\mathrm{T}^{-1} & 0 & 0 \\ \frac{1}{2}\mathrm{Q} & 0 & 0 \\ 0 & 0 & -\mathrm{N} \end{bmatrix} \qquad (34.a)$$

The dynamics model is then

$$\dot{\mathbf{x}}_s = \mathrm{F}_s\mathbf{x}_s + \mathbf{f} + \mathbf{w}_s \qquad (34.b)$$

where

$$\mathbf{x}_s^{\mathrm{T}} = [\underline{\omega}_s^{\mathrm{T}} \,|\, \mathbf{q}_s^{\mathrm{T}} \,|\, \mathbf{v}_1^{\mathrm{T}}] \qquad (34.c)$$

f is as before and

$$\mathbf{w}_s^{\mathrm{T}} = [\mathbf{w}_s^{\mathrm{T}} \,|\, \mathbf{0}^{\mathrm{T}} \,|\, \boldsymbol{\mu}_1^{\mathrm{T}}] \qquad (34.d)$$

The covariance matrix of $\mathbf{w}_S$ has to be computed[15] and tuned. When the quaternion measurements are used to update the filter every second there is almost no visible difference between the use of the elaborate rotational dynamics model and the simplified Markov model. However if the updates occur at longer intervals there is a remarkable difference between the two cases. Fig. 6 presents the angular-rate estimation error when the elaborate angular dynamics is used and the PSELIKA filter, which is used to estimate the rates, is updated at an arbitrarily 30 second rate. The FM computation of the error presented in Fig. 6 results in $\mathrm{FM}(6) = 1.7975 \cdot 10^{-3}$ deg/sec. When the elaborate model is replaced by the Markov model, the error in the resulting estimated rate is unacceptable. This is seen in Fig. 7 where the angular-rate estimation errors for this case are presented. This is also indicated by the large FM of this case where $\mathrm{FM}(7) = 3.9136 \cdot 10^{-2}$ deg/sec. It should be noted that in the computation of FM(6) and FM(7) we set $t_0 = 200\,\text{sec}$. Again, this was done in order to avoid the transients. In summary, the simplified model was shown to be unacceptable.

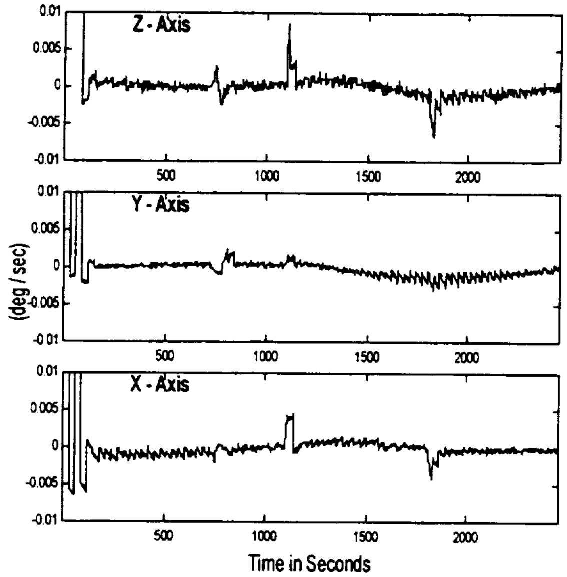


Fig. 6: The Estimated Angular-Rate Error After Applying the PSELIKA Filter to the Augmented Model with Sparse Measurements.

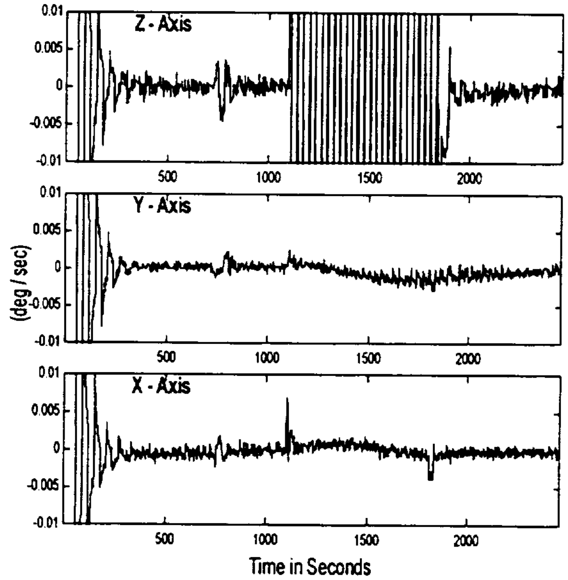


Fig. 7: The Estimated Angular-Rate Error After Applying the PSELIKA Filter to the Simplified Model with Sparse Measurements.

VII. QUATERNION NORMALIZATION

Since quaternions are inherently normal, the quaternion which is estimated using the above algorithms has to be normal; however, these estimation algorithms do not assure normalization; therefore, occasionally, the estimated quaternion has to be normalized. Several algorithms were suggested for it in the past[16,17] which were compatible with the KF estimator. The accuracy achieved when using those algorithms was about the same for all of them. In this work we chose to apply the Magnitude Pseudo-Measurement (MPM) normalization algorithm[17] for the ease of its implementation. This algorithm is presented next.

The states which constitute the four elements of the quaternion are x_4, x_5, x_6 and x_7 therefore the sum $\hat{x}_4^2 + \hat{x}_5^2 + \hat{x}_6^2 + \hat{x}_7^2$ has to be equal to 1. In order to assure it we assume the existence of a "magnitude measuring device" that "measures" 1; that is,

$$z_{\text{norm},k+1} = 1 \qquad (35.a)$$

On the other hand we assume that the corresponding measurement model is

$$z_{\text{norm},k+1} = [0,0,0,\hat{x}_{4,k+1},\hat{x}_{5,k+1},\hat{x}_{6,k+1},\hat{x}_{7,k+1},0,0,0,0]\,\mathbf{x} + v_{\text{norm},k+1} \qquad (35.b)$$

which can be written in the form

$$z_{\text{norm},k+1} = H_{\text{norm}}\,\mathbf{x} + v_{\text{norm},k+1} \qquad (35.c)$$

where, obviously

$$H_{\text{norm}} = [0,0,0,\hat{x}_{4,k+1},\hat{x}_{5,k+1},\hat{x}_{6,k+1},\hat{x}_{7,k+1},0,0,0,0] \qquad (35.d)$$

It is possible now to perform an ordinary measurement update where the filter is fed with the "measurement" 1, and where the measurement matrix is given in Eq. (35.d). The value of $r_{\text{norm},k+1}$, the variance of the "measurement" error $v_{\text{norm},k+1}$, can be adjusted to yield satisfactory results. We note that indeed this algorithm forces normality on the estimated quaternion without violating the KF rules.

VIII. DELAYED QUATERNION MEASUREMENTS

The device that yields the quaternion measurements[9] computes the quaternion after a star search, therefore the quaternion is obtained with a time delay. Fig. 8 presents the time points where measurement updates take place and the size of the delay time, Δ, which, for reasons explained later, is divided into subintervals of length δ. The filter has to supply the best estimate of the angular-rate to the SC attitude control system (ACS) at points t_k, t_{k+1}, etc. However, the measurement which is obtained at t_{k+1} is of the quaternion that existed at $t_{meas,k+1}$; that is, at time point $t_{k+1} - \Delta$. There are several ways to process the delayed measurement in order to obtain an improved estimate at t_{k+1}. In the ensuing we present two algorithms. According to the first algorithm, which we name *Updating Before Propagating*, we perform a measurement update of the filter at time point $t_{meas,k+1}$, when the real time is already t_{k+1}, and then propagate the outcome to time point measurement where the information is passed on to the ACS. It is also possible to first propagate the state estimate and covariance matrix (when PSELIKA is used) to the time point t_{k+1}, project the measurement too to this time point, and only then perform a measurement update. We name this second algorithm *Updating After Propagating*. Both algorithms are further explained in the following.

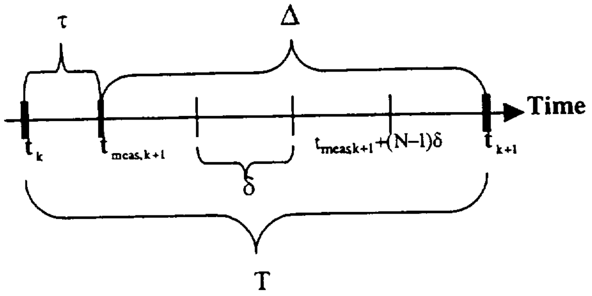


Fig. 8: The Relative Location of the Delayed Quaternion Measurement Along the Time Line.

VIII.1 Updating Before Propagating

The sequence of events concerning the propagation and the updating of the state estimate when using this algorithm is presented in Fig. 9. Since τ is known (see Fig. 8 for the meaning of τ), it is possible to propagate the state vector and the covariance matrix from time t_k to $t_{meas,k+1}$ and stop the propagation there until the real time reaches t_{k+1}. The state estimate is propagated from t_k to $t_{meas,k+1}$ using the following discretized version of Eqs. (31) and (32)

$$\begin{bmatrix} \mathbf{e}^{F(\hat{\underline{\omega}})\tau} & 0 & 0 \\ \frac{1}{2}QF^{-1}(\hat{\underline{\omega}})[\mathbf{e}^{F(\hat{\underline{\omega}})\tau} - I] & I & 0 \\ 0 & 0 & \mathbf{e}^{-N\tau} \end{bmatrix} \begin{bmatrix} \hat{\underline{\omega}}_{k+1-\Delta}(-) \\ \hat{\mathbf{q}}_{k+1-\Delta}(-) \\ \hat{\mathbf{v}}_{1,k+1-\Delta}(-) \end{bmatrix} = \begin{bmatrix} \hat{\underline{\omega}}_{k}(-) \\ \hat{\mathbf{q}}_{k}(-) \\ \hat{\mathbf{v}}_{1,k}(-) \end{bmatrix}$$

$$+ \begin{bmatrix} \mathbf{u}_k \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \qquad (36.a)$$

and, when using PSELIKA, the covariance matrix is propagated using Eq. (11.b) noting that

$$A_k = \begin{bmatrix} \mathbf{e}^{F(\hat{\underline{\omega}})\tau} & 0 & 0 \\ \frac{1}{2}QF^{-1}(\hat{\underline{\omega}})[\mathbf{e}^{F(\hat{\underline{\omega}})\tau} - I] & I & 0 \\ 0 & 0 & \mathbf{e}^{-N\tau} \end{bmatrix} \qquad (36.b)$$

Note that when the SDARE filter is used, there is no covariance matrix propagation. (One needs not worry about a possible singularity of $F(\underline{\omega})$ which may appear in the 2,1 element of the discretized dynamics matrix in Eqs. (36) because $F^{-1}(\underline{\omega})$ is included in this element only in order to enable the expression of the term in a closed form. If this term is expressed in a power series form, the inverse of $F(\underline{\omega})$ is canceled out).

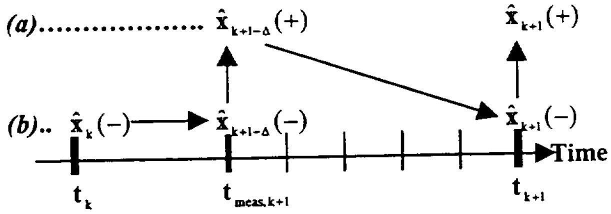


Fig. 9: The Evolution of the State Estimate Over One Time Cycle.
(a) updated vectors. (b) propagated vectors.

When the real time reaches t_{k+1} the measurement that was performed at time $t_{meas,k+1}$ is available, therefore it is possible to use the state and covariance matrix (when PSELIKA is used), which have been propagated to $t_{meas,k+1}$, and the new measurement to perform a measurement update at time $t_{meas,k+1}$ which yields

$$\hat{\mathbf{x}}_{k+1-\Delta}(+) = [\hat{\underline{\omega}}^{T}_{k+1-\Delta}(+) \,|\, \hat{\mathbf{q}}^{T}_{k+1-\Delta}(+) \,|\, \hat{\mathbf{v}}^{T}_{1,k+1-\Delta}(+)]^{T} \tag{37}$$

Following the update the state estimate and covariance matrix can be propagated to the real time, t_{k+1}, for use as an input to the ACS. The state estimate is propagated N times using the small step size, δ (see Fig. 7), and when PSELIKA is used, the covariance matrix too is propagated N times up to t_{k+1}. This propagation results in $\hat{\mathbf{x}}_{k+1}(-)$, and when PSELIKA is used, also in $P_{k+1}(-)$. Note that the measurement update and the subsequent propagation take place only after the real time reaches t_{k+1}, and since these operations take time, they cause a delay in the transfer of data to the ACS which is supposed to occur at precisely t_{k+1}. We assume, however, that the computation time and the resulting delay are negligible. If this is not the case there are ways to compensate for this delay.

VIII.2 Updating After Propagating

The sequence of events concerning the propagation and the updating of the state estimate when using this algorithm is presented in Fig. 9. Here first the state estimate is propagated to time point t_{k+1}, and if PSELIKA is used the covariance matrix is also propagated. Then when the real time reaches t_{k+1} the measurement of $\mathbf{q}_{k+1-\Delta}$ is available. It is also propagated to t_{k+1} (not shown in Fig. 9) and an update is performed there.

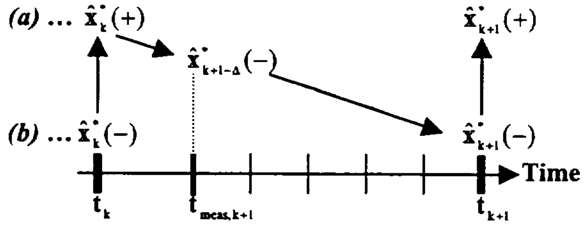

(a) ... $\hat{\mathbf{x}}^{*}_{k}(+)$
$\hat{\mathbf{x}}^{*}_{k+1}(+)$
$\hat{\mathbf{x}}^{*}_{k+1-\Delta}(-)$
(b) ... $\hat{\mathbf{x}}^{*}_{k}(-)$
$\hat{\mathbf{x}}^{*}_{k+1}(-)$
Time
t_k
$t_{meas,k+1}$
t_{k+1}

Fig. 10: The Evolution of the State Estimate Over One Time Cycle. ***(a) updated vectors. (b) propagated vectors.***

For reasons which will be clear later define a state vector $\mathbf{x}^{*}$ as follows

$$\mathbf{x}^{*}_{k} = [\underline{\omega}^{T}_{k} \,|\, \mathbf{q}^{T}_{k} \,|\, \mathbf{v}^{T}_{1,k-\Delta}]^{T} \tag{38}$$

As will be shown later, the measurement update at time t_k yields

$$\hat{\mathbf{x}}^{*}_{k}(+) = [\hat{\underline{\omega}}^{T}_{k}(+) \,|\, \hat{\mathbf{q}}^{T}_{k}(+) \,|\, \hat{\mathbf{v}}^{T}_{1,k-\Delta}(+)]^{T} \tag{39}$$

This state estimate as well as the covariance (when PSELIKA is used) are propagated first by τ to $t_{meas,k+1}$. The propagation is done like in the previous algorithm (see Eqs. (36)). (Note that although $\mathbf{v}_{1,k-\Delta}$ is propagated τ seconds, it does not reach the time point $t_{meas,k+1}$). Next the propagation continues from time point $t_{meas,k+1}$ to t_{k+1} by the small time steps δ, and at time points $t_{meas,k+1} + i \cdot \delta$ for $i = 0, 1, N\text{-}1$, $\underline{\omega}$ is recorded as $\underline{\omega}(t_{i\cdot\delta})$. Next the measurement at time point $t_{meas,k+1}$ is projected to time point t_{k+1}. Observe that in view of Eq. (33.a) one can write

$$\mathbf{q}_{meas,k+1} = \mathbf{q}_{k+1-\Delta} + \mathbf{v}_{1,k+1-\Delta} + \mathbf{v}_{2,k+1-\Delta} \tag{40}$$

where $\mathbf{q}_{meas,k+1}$ is *the measurement obtained at time* t_{k+1} *of* $\mathbf{q}$ *at time* $t_{meas,k+1}$; that is, of $\mathbf{q}_{k+1-\Delta}$. The vectors $\mathbf{v}_{1,k+1-\Delta}$ and $\mathbf{v}_{2,k+1-\Delta}$ are, respectively, the measurement-noise vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ which exist at time $t_{k+1} - \Delta$. Denote the transition matrix of $\mathbf{q}$ from $t_{meas,k+1}$ to t_{k+1} by $\Phi_q(\Delta)$, then, using the recorded $\omega(t_{i\cdot\delta})$, the latter can be computed as

$$\Phi_q(\Delta) = \prod_{i=0}^{i=N-1} e^{\frac{1}{2}\Omega(t_{i\cdot\delta})} \tag{41}$$

and the measurement can be projected forward from $t_{meas,k+1}$ to t_{k+1}, as follows

$$\mathbf{q}_{m,k+1} = \Phi_q(\Delta)\, \mathbf{q}_{meas,k+1} \tag{42}$$

From Eqs. (40) and (42) it is clear that

$$\mathbf{q}_{m,k+1} = \Phi_q(\Delta)\, \mathbf{q}_{k+1-\Delta} + \Phi_q(\Delta)\, \mathbf{v}_{1,k+1-\Delta} + \Phi_q(\Delta)\, \mathbf{v}_{2,k+1-\Delta} \tag{43}$$

When examining the terms on the right hand side of Eq. (43) one realizes that

$$\Phi_q(\Delta)\,\mathbf{q}_{k+1-\Delta} = \mathbf{q}_{k+1} \qquad (44)$$

Since the differential equation which describes the propagation of $\mathbf{v}_1$ is non homogeneous, then

$$\Phi_q(\Delta)\,\mathbf{v}_{1,k+1-\Delta} \neq \mathbf{v}_{1,k+1} \qquad (45)$$

therefore Eqs. (43) and (44) are written as follows

$$\mathbf{q}_{m,k+1} = [0_{4x3} \mid I_{4x4} \mid \Phi_q(\Delta)\,] \begin{bmatrix} \underline{\omega}_{k+1} \\ \mathbf{q}_{k+1} \\ \mathbf{v}_{1,k+1-\Delta} \end{bmatrix} + \Phi_q(\Delta)\,\mathbf{v}_{2,k+1-\Delta} \qquad (46)$$

Define

$$\mathbf{z}_{m,k+1} = \mathbf{q}_{m,k+1} \qquad (47.a)$$

$$\mathbf{x}^{\bullet}_{k+1} = [\underline{\omega}^{T}_{k+1} \mid \mathbf{q}^{T}_{k+1} \mid \mathbf{v}^{T}_{1,k+1-\Delta}]^{T} \qquad (47.b)$$

$$H_{m,k+1} = [0_{4x3} \mid I_{4x4} \mid \Phi(\Delta)\,] \qquad (47.c)$$

and

$$\underline{\nu}_{k+1} = \Phi(\Delta)\,\mathbf{v}_{2,k+1-\Delta} \qquad (47.d)$$

then Eq. (46) can be written as

$$\mathbf{z}_{m,k+1} = H_{m,k+1}\mathbf{x}^{\bullet}_{k+1} + \underline{\nu}_{k+1} \qquad (48)$$

According to Eq. (47.d), R_{ν}, the covariance matrix of $\underline{\nu}_{k+1}$, is computed as follows

$$R_{\nu} = \Phi_q(\Delta)R_{2,k+1-\Delta}\Phi^{T}_q(\Delta) \qquad (49)$$

where $R_{2,k+1-\Delta}$ is the covariance matrix of $\mathbf{v}_2$ evaluated at $t_{meas,k+1}$. Using Eqs. (47.a), (47.c) and (49) we can now perform a measurements update using either PSELIKA or SDARE. This yields

$$\hat{\mathbf{x}}^{\bullet}_{k+1}(+) = [\hat{\underline{\omega}}^{T}_{k+1}(+) \mid \hat{\mathbf{q}}^{T}_{k+1}(+) \mid \hat{\mathbf{v}}^{T}_{1,k+1-\Delta}(+)]^{T} \qquad (50)$$

and $\hat{\underline{\omega}}_{k+1}(+)$ can now be fed into the ACS as required. It is clear now why in Eq. (38) the new vector $\mathbf{x}^{\bullet}$ was defined. Note that $\mathbf{x}^{\bullet}$ contains vectors of different times, but this poses no problem. In fact this situation occurs in certain smoothing algorithms. One has to be aware though of the fact that the off diagonal elements in the covariance matrix which relate $\mathbf{v}_1$ to $\underline{\omega}$ and to $\mathbf{q}$, yields the covariance of the estimation error of vectors at two different time points.

IX. CONCLUSIONS

In this paper we examined algorithms for estimating the angular-rate vector of satellites using quaternion measurements without differentiation. The notion examined in this work is based on the ability to obtain quaternion measurements directly from star tracker(s). For the sake of comparison we also examined the approach of extracting the angular-rate from quaternion differentiation. Both approaches utilize a Kalman filter. In fact two filters were examined. One was the PSEudo-Linear KAlman (PSELIKA) filter and the other was a special Kalman filter which was based on the use of the solution of the State Dependent Algebraic Riccati Equation (SDARE) in order to compute the Kalman gain matrix and thus eliminate the need to propagate and update the filter covariance matrix. The two filters relied on the ability to decompose the non-linear rate dependent part of the rotational dynamics equation of a rigid body into a product of an angular-rate dependent matrix and the angular-rate vector itself. This non-unique decomposition enabled the treatment of the nonlinear spacecraft dynamics model as a linear one and, consequently, the application of the PSELIKA filter. It also enabled the application of the SDARE filter.

When using the quaternion measurements to obtain angular-rate without differentiation, the kinematics equation of the quaternion has to be incorporated into the filter dynamics model. This can be done in two ways. It was shown that only one way can be used because only this way yields an observable system.

Real spacecraft data was used to test the suggested algorithms. As expected, when rate determination was based on quaternion differentiation, the resulting angular-rate was noisy. When either one of the filters was used, the noise was suppressed without causing delays in the estimated angular-rate components.

The replacement of the elaborate rotational dynamics by a simple first order Markov model was also examined. It was found that while the use of such a simple model was sufficient when frequent measurement updates were possible, it was totally inadequate when only sparse quaternion measurements were available.

It was explained that the device that yields the quaternion measurements computes the quaternion after a star search, therefore the quaternion is obtained with a time delay. However the filter has to supply the best estimate of the angular-rate to the SC attitude control system on time. In this work two algorithms were presented to overcome the delay problem. According to the first algorithm, which we named Updating Before Propagating, a measurement update of the filter is performed at the time when the measurement is obtained and then propagated to the time point where the information is passed on to the attitude control system. In the second algorithm the state estimate, the covariance matrix (when PSELIKA is used), and the measurement are first propagated to the time point where the angular-rate has to be passed on to the attitude control system, and only then a measurement update is performed. We named this second algorithm Updating After Propagating.

This paper presents results which indicate that a high performance attitude and rate estimation algorithm can be implemented using star trackers and without the use of costly high performance gyros. These results should be considered for all future missions with high performance rate estimation requirements and for those missions which are looking for a high performance gyro-less backup.

REFERENCES

1. Harman, R.R., and Bar-Itzhack, I.Y., "Angular-rate Computation Using Attitude Differentiation," Technical Report No. TAE-881, Faculty of Aerospace Eng., Technion-Israel Institute of Technology, Haifa 32000, Israel, Dec. 97.

2. Harman, R.R., and Bar-Itzhack, I.Y., "The Use of Pseudo-Linear and SDARE Filtering for Satellite Angular-Rate Estimation," AIAA Guidance, Navigation, and Control Conf., Boston, MA, August 10-12, 1997.

3. Azor, R., Bar-Itzhack, I. Y., and Harman, R. R., "Satellite Angular-rate Estimation from Vector Measurements," *J. of Guidance, Control, and Dynamics*, Vol. 21, No. 3, May-June, 1998, pp. 450-457.

4. Natanson, G., "A Deterministic Method for Estimating Attitude from Magnetometer Data Only," Proceedings of the 43rd IFA Congress, Paper No. IFA-92-0036, Washington, DC, Aug. 28-Sept. 5, 1992.

5. Challa, M., Natanson, G., Deutschmann, J., and Galal, K., "A PC-Based Magnetometer-Only Attitude and Rate Determination System for Gyroless Spacecraft," Proceedings of the Flight Mechanics/Estimation Theory Symposium, NASA-Goddard Space Flight Center, April 1995.

6. Challa, M., Kotaru, S., and Natanson, G., "Magnetometer-Only Attitude and Rate Estimates During the Earth Radiation Budget Satellite 1987 Control Anomaly," AIAA Guidance, Navigation, and Control Conf., New Orleans, LA, August 11-13, 1997.

7. Oshman, Y., and Markley, F. L., "Sequential Attitude and Attitude-Rate Estimation Using Integrated-Rate Parameters," to be published in the *J. of Guidance, Control, and Dynamics*.

8. Dellus, F., "Estimation of Satellite Angular Velocity Using Sequential Measurements of a Single Inertial Vector," M.Sc. Thesis, Faculty of Aerospace Eng., Technion-Israel Institute of Technology, June 1998. (In Hebrew).

9. Van Bezooijen, R.W.H., "AST Capabilities," Lockheed Martin Advanced Technology Center, Palo Alto, CA 95304-1191. (Slide presentation).

10. Wertz, J. R., (Ed.), *Spacecraft Attitude Dynamics and Control*, Reidel Publishing Co., Dordrecht, Holland, 1999.

11. Cloutier, J.R., D'Souza, C.N., and Mracek, C.P., "Nonlinear Regulation and Nonlinear H_∞ Control Via the State-Dependent Riccati Equation Technique: Part 1, Theory," The First International Conf. on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, FL, May 1996.

12. Cloutier, J.R., D'Souza, C.N., and Mracek, C.P., "Nonlinear Regulation and Nonlinear H_∞ Control Via the State-Dependent Riccati Equation Technique: Part 2, Examples," The First International Conf. on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, FL, May 1996.

13. Pappano, V., and Friedland, B., "SDRE Observer for an Induction Machine," The American Control Conf., Albuquerque, NM, June 1997.

14. Mracek, C.P., Cloutier, J.R., and D'Souza, C.N., "A New Technique for Nonlinear Estimation," IEEE Conf. on Control Applications, Dearborn, MI, September 1996.

15. Gelb, A., (ed.), *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

16. Bar-Itzhack, I.Y., and Oshman, Y., "Recursive Attitude Determination from Vector Observations: Quaternion Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-21, Jan. 1985, pp. 128-136.

17. Deutschmann, J.K., Bar-Itzhack, I.Y., and Markley, F.L., "Quaternion Normalization in Additive EKF for Spacecraft Attitude Determination," AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, August 1991.

ATTITUDE DETERMINATION USING TWO VECTOR MEASUREMENTS

F. Landis Markley
Guidance, Navigation, and Control Systems Engineering Branch, Code 571
NASA's Goddard Space Flight Center, Greenbelt, MD 20771

ABSTRACT

Many spacecraft attitude determination methods use exactly two vector measurements. The two vectors are typically the unit vector to the Sun and the Earth's magnetic field vector for coarse "sun-mag" attitude determination or unit vectors to two stars tracked by two star trackers for fine attitude determination. TRIAD, the earliest published algorithm for determining spacecraft attitude from two vector measurements, has been widely used in both ground-based and onboard attitude determination. Later attitude determination methods have been based on Wahba's optimality criterion for *n* arbitrarily weighted observations. The solution of Wahba's problem is somewhat difficult in the general case, but there is a simple closed-form solution in the two-observation case. This solution reduces to the TRIAD solution for certain choices of measurement weights. This paper presents and compares these algorithms as well as sub-optimal algorithms proposed by Bar-Itzhack, Harman, and Reynolds. Some new results will be presented, but the paper is primarily a review and tutorial.

INTRODUCTION

Suppose that we have measured two unit vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ in the spacecraft body frame. These can be the unit vectors to an observed object like a star or the Sun, or some ambient vector field such as the Earth's magnetic field. We consider only unit vectors because the length of the vector has no information relevant to attitude determination. Each of these unit vectors thus contains two independent scalar pieces of attitude information. The spacecraft attitude is represented by a 3×3 orthogonal matrix *A*, *i.e.* $A^T A = I$, the 3×3 identity matrix. The attitude matrix must also be proper, *i.e.*, it must have unit determinant, so it is an element of the three-parameter group SO(3). Euler's Theorem states that the most general motion of a rigid body with one fixed point is a rotation about some axis. This shows explicitly that SO(3) is a three-parameter group, since the three parameters can be taken as the rotation angle and two parameters specifying a unit vector along the rotation axis. Thus two unit vector measurements determine the attitude matrix, in general; in fact they overdetermine it.

It is also necessary to know the components of the two measured vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ in some reference frame. The reference frame is usually taken to be an inertial frame, but this is not necessary. One can use a rotating frame such as the frame referenced to the orbit normal vector and the local vertical. The attitude matrix to be determined is the matrix that rotates vectors from the reference frame to the spacecraft body frame. Thus we would like to find an attitude matrix such that

$$A\mathbf{r}_1 = \mathbf{b}_1 \tag{1a}$$

and

$$A\mathbf{r}_2 = \mathbf{b}_2 . \tag{1b}$$

This is not possible in general, however, for equation (1) implies that

$$\mathbf{b}_1 \cdot \mathbf{b}_2 = (A\mathbf{r}_1)\cdot(A\mathbf{r}_2) = \mathbf{r}_1^T A^T A \mathbf{r}_2 = \mathbf{r}_1^T \mathbf{r}_2 = \mathbf{r}_1 \cdot \mathbf{r}_2 . \tag{2}$$

This equality is true for error-free measurements, but is not generally true in the presence of measurement errors. It will be seen in the following that all reasonable two-vector attitude determination schemes give the same estimate when equation (2) is valid.

It is clear from simple counting arguments that the two independent scalar pieces of information contained in a single vector measurement cannot determine the attitude uniquely. More concretely, if the attitude matrix *A* obeys equation (1a), then so does the matrix $R(\mathbf{b}_1,\phi_b)AR(\mathbf{r}_1,\phi_r)$, for any ϕ_b and ϕ_r, where $R(\mathbf{e},\phi)$ denotes a rotation by angle ϕ about the axis $\mathbf{e}$. This line of argument also makes it clear that the attitude matrix is not uniquely determined if either the pair $\mathbf{b}_1$ and $\mathbf{b}_2$ or the pair $\mathbf{r}_1$ and $\mathbf{r}_2$ are parallel or antiparallel.

The earliest published algorithm for determining spacecraft attitude from two vector measurements was the TRIAD algorithm[1,2]. This algorithm has been widely used in both ground-based and onboard[3] attitude determination. The two vectors are typically the unit vector to the Sun and the Earth's magnetic field vector for coarse "sun-mag" attitude determination or unit vectors to two stars tracked by two star trackers for fine attitude determination. Recent developments in star tracker technology have produced star trackers that can track 5, 6, or even 50 stars at a time. For attitude determination using more than two vectors, optimal estimators based on a loss function introduced by Wahba are appropriate[4]. However, Bronzenac and Bender have shown that the n vectors from a small-field-of-view star tracker can be replaced by an average vector without significant loss of precision[5]. With this approximation, the two star tracker case, even with multiple stars tracked in each star tracker, can be treated as a two-vector-measurement problem.

With this motivation, we survey solutions to the two-vector measurement problem, beginning with TRIAD. We then consider the optimal solution of Wahba's problem. After this, we look at sub-optimal algorithms have been proposed by Bar-Itzhack and Harman[6] and by Reynolds[7,8]. We compare the various algorithms for both accuracy and computational effort, and finally present conclusions.

TRIAD

The TRIAD algorithm, introduced by Black in 1964[1,2], is based on the following idea. If we have an orthogonal right-handed triad of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ in the reference frame, and a corresponding triad $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ in the spacecraft body frame, the the attitude matrix

$$A = [\mathbf{w}_1 \vdots \mathbf{w}_2 \vdots \mathbf{w}_3][\mathbf{v}_1 \vdots \mathbf{v}_2 \vdots \mathbf{v}_3]^T = \mathbf{w}_1\mathbf{v}_1^T + \mathbf{w}_2\mathbf{v}_2^T + \mathbf{w}_3\mathbf{v}_3^T \tag{3}$$

will transform the $\mathbf{v}_i$ to the $\mathbf{w}_i$ by

$$A\mathbf{v}_i = \mathbf{w}_i, \; i = 1, 2, 3. \tag{4}$$

The TRIAD algorithm forms the triad $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ from $\mathbf{r}_1$ and $\mathbf{r}_2$, and the triad $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ from $\mathbf{b}_1$ and $\mathbf{b}_2$. Incidentally, TRIAD can be considered either as the word "triad" or as an acronym for "TRIaxial Attitude Determination." The triads can be formed in three convenient ways. First, it is useful to define the normalized cross products

$$\mathbf{r}_3 \equiv (\mathbf{r}_1 \times \mathbf{r}_2)/|\mathbf{r}_1 \times \mathbf{r}_2| \tag{5a}$$

and

$$\mathbf{b}_3 \equiv (\mathbf{b}_1 \times \mathbf{b}_2)/|\mathbf{b}_1 \times \mathbf{b}_2|. \tag{5b}$$

We note that $\mathbf{r}_3$ or $\mathbf{b}_3$ is undefined if the reference vectors or the observed vectors, respectively, are parallel or antiparallel. This is the case noted above in which there is insufficient information to determine the attitude uniquely. If this is not the case, two of the TRIAD attitude estimates are

$$A_{T1} \equiv \mathbf{b}_1\mathbf{r}_1^T + \mathbf{b}_3\mathbf{r}_3^T + (\mathbf{b}_1 \times \mathbf{b}_3)(\mathbf{r}_1 \times \mathbf{r}_3)^T \tag{6}$$

and

$$A_{T2} \equiv \mathbf{b}_2\mathbf{r}_2^T + \mathbf{b}_3\mathbf{r}_3^T + (\mathbf{b}_2 \times \mathbf{b}_3)(\mathbf{r}_2 \times \mathbf{r}_3)^T. \tag{7}$$

These estimates treat the two measurements unsymmetrically. In fact $A_{T1}\mathbf{r}_1 = \mathbf{b}_1$ and $A_{T2}\mathbf{r}_2 = \mathbf{b}_2$, but

$$A_{T1}\mathbf{r}_2 \equiv \mathbf{b}_1(\mathbf{r}_1 \cdot \mathbf{r}_2) + (\mathbf{b}_1 \times \mathbf{b}_3)[(\mathbf{r}_1 \times \mathbf{r}_3)\cdot \mathbf{r}_2] = (\mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{b}_1 + [\mathbf{b}_2 - (\mathbf{b}_1 \cdot \mathbf{b}_2)\mathbf{b}_1]|\mathbf{r}_1 \times \mathbf{r}_2|/|\mathbf{b}_1 \times \mathbf{b}_2| \tag{8}$$

and

$$A_{T2}\mathbf{r}_1 \equiv \mathbf{b}_2(\mathbf{r}_1 \cdot \mathbf{r}_2) + (\mathbf{b}_2 \times \mathbf{b}_3)[(\mathbf{r}_2 \times \mathbf{r}_3)\cdot \mathbf{r}_1] = (\mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{b}_2 + [\mathbf{b}_1 - (\mathbf{b}_1 \cdot \mathbf{b}_2)\mathbf{b}_2]|\mathbf{r}_1 \times \mathbf{r}_2|/|\mathbf{b}_1 \times \mathbf{b}_2|. \tag{9}$$

Thus the estimate A_{T1} emphasizes the first measurement and A_{T2} emphasizes the second. It's not difficult to see, though, that both A_{T1} and A_{T2} satisfy equations (1a) and (1b) if $\mathbf{b}_1 \cdot \mathbf{b}_2 = \mathbf{r}_1 \cdot \mathbf{r}_2$.

The third form of TRIAD treats the two measurements symmetrically. We define the unit vectors

$$\mathbf{r}_+ \equiv (\mathbf{r}_2 + \mathbf{r}_1)/|\mathbf{r}_2 + \mathbf{r}_1| = (\mathbf{r}_2 + \mathbf{r}_1)/\sqrt{2(1 + \mathbf{r}_1 \cdot \mathbf{r}_2)}, \tag{10}$$

$$\mathbf{r}_- \equiv (\mathbf{r}_2 - \mathbf{r}_1)/|\mathbf{r}_2 - \mathbf{r}_1| = (\mathbf{r}_2 - \mathbf{r}_1)/\sqrt{2(1 - \mathbf{r}_1 \cdot \mathbf{r}_2)}, \tag{11}$$

and $\mathbf{b}_+$ and $\mathbf{b}_-$ similarly. It is easy to see that $\mathbf{r}_+$ is perpendicular to $\mathbf{r}_-$, $\mathbf{b}_+$ is perpendicular to $\mathbf{b}_-$, and also that $\mathbf{r}_3 = \mathbf{r}_+ \times \mathbf{r}_-$ and $\mathbf{b}_3 = \mathbf{b}_+ \times \mathbf{b}_-$. Thus $\{\mathbf{r}_+, \mathbf{r}_-, \mathbf{r}_3\}$ and $\{\mathbf{b}_+, \mathbf{b}_-, \mathbf{b}_3\}$ are orthogonal triads, and the third TRIAD estimate is given by

$$A_{T3} \equiv \mathbf{b}_+\mathbf{r}_+^T + \mathbf{b}_-\mathbf{r}_-^T + (\mathbf{b}_+ \times \mathbf{b}_-)(\mathbf{r}_+ \times \mathbf{r}_-)^T. \tag{12}$$

This estimate treats the two observations symmetrically, and gives $A_{T3}\mathbf{r}_+ = \mathbf{b}_+$ and $A_{T3}\mathbf{r}_- = \mathbf{b}_-$, but

$$A_{T3}\mathbf{r}_1 \equiv \mathbf{b}_+(\mathbf{r}_+\cdot\mathbf{r}_1)+\mathbf{b}_-(\mathbf{r}_-\cdot\mathbf{r}_1) = \frac{1}{2}\left[\sqrt{\frac{1+\mathbf{r}_1\cdot\mathbf{r}_2}{1+\mathbf{b}_1\cdot\mathbf{b}_2}}\,(\mathbf{b}_1+\mathbf{b}_2)+\sqrt{\frac{1-\mathbf{r}_1\cdot\mathbf{r}_2}{1-\mathbf{b}_1\cdot\mathbf{b}_2}}\,(\mathbf{b}_1-\mathbf{b}_2)\right] \tag{13}$$

and

$$A_{T3}\mathbf{r}_2 \equiv \mathbf{b}_+(\mathbf{r}_+\cdot\mathbf{r}_2)+\mathbf{b}_-(\mathbf{r}_-\cdot\mathbf{r}_2) = \frac{1}{2}\left[\sqrt{\frac{1+\mathbf{r}_1\cdot\mathbf{r}_2}{1+\mathbf{b}_1\cdot\mathbf{b}_2}}\,(\mathbf{b}_1+\mathbf{b}_2)-\sqrt{\frac{1-\mathbf{r}_1\cdot\mathbf{r}_2}{1-\mathbf{b}_1\cdot\mathbf{b}_2}}\,(\mathbf{b}_1-\mathbf{b}_2)\right]. \tag{14}$$

Again, it's not difficult to see that A_{T3} satisfies equations (1a) and (1b) if $\mathbf{b}_1\cdot\mathbf{b}_2 = \mathbf{r}_1\cdot\mathbf{r}_2$.

All three TRIAD estimates satisfy $A_{Ti}\mathbf{r}_3 = \mathbf{b}_3$, for i = 1, 2, 3. From this and the above observations, it is clear that A_{T1}, A_{T2}, and A_{T3} give identical estimates if equation (2) is valid, since they provide the same mapping of a basis $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$ in the reference frame to a basis $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ in the spacecraft body frame.

THE OPTIMAL SOLUTION

In 1965, Grace Wahba, then a graduate student at Stanford University on a summer job with IBM, proposed the following problem[4]: Find the orthogonal matrix A with determinant +1 that minimizes the loss function

$$L(A) \equiv \tfrac{1}{2}\sum\nolimits_i a_i|\mathbf{b}_i - A\mathbf{r}_i|^2 . \tag{15}$$

where $\{\mathbf{b}_i\}$ is a set of n unit vectors measured in a spacecraft's body frame, $\{\mathbf{r}_i\}$ are the corresponding unit vectors in a reference frame, and $\{a_i\}$ are non-negative weights. We can rewrite equation (15), using the invariance of the trace under cyclic permutations, as

$$L(A) = \tfrac{1}{2}\sum\nolimits_i a_i\left(|\mathbf{b}_i|^2+|\mathbf{r}_i|^2\right) - \sum\nolimits_i a_i\mathbf{b}_i^T A\mathbf{r}_i = \left(\sum\nolimits_i a_i\right) - \operatorname{trace}(AB^T), \tag{16}$$

where

$$B \equiv \sum\nolimits_i a_i\mathbf{b}_i\mathbf{r}_i^T . \tag{17}$$

It is obvious that the attitude matrix that minimizes the loss function is the proper orthogonal matrix that maximizes trace(AB^T). Almost all solutions of Wahba's problem are based on this observation. The original solutions solved for the attitude matrix A directly, but most practical applications have been based on Davenport's q-method[2,9], which solves for the attitude quaternion[10,11]. Shuster's QUEST algorithm, in particular, has been widely used[12]. Shuster showed a simplification in the two-observation Wahba problem, but the first explicit closed-form solution was presented in reference 13.

We begin by noting that the matrix B has the singular value decomposition[14-16]

$$B = USV^T, \tag{18}$$

where U and V are orthogonal matrices, and S is diagonal;

$$S = \operatorname{diag}(s_1, s_2, s_3), \tag{19}$$

with

$$s_1 \geq s_2 \geq s_3 \geq 0. \tag{20}$$

In the two-observation case, it is clear from equation (17) that B has rank at most 2, and therefore

$$\det B = s_1 s_2 s_3 = 0. \tag{21}$$

Equations (20) and (21) show that

$$s_3 = 0 \tag{22}$$

in the two-observation case. We shall take advantage of some resulting simplifications in this case. The general n–observation case is treated in references 13 and 14.

Since $s_3 = 0$, we are free to choose the sign of the last column of U and of V so that both of these matrices have positive determinants. We shall assume that this is the case. Now

$$\operatorname{trace}(AB^T) = \operatorname{trace}(AVSU^T) = \operatorname{trace}(WS) = s_1W_{11} + s_2W_{22}, \tag{23}$$

where we have again used the invariance of the trace under cyclic permutations, and

$$W \equiv U^T A V. \tag{24}$$

Now using the Euler axis/angle parameterization for $W = R(\mathbf{e},\phi)$ gives[10,11]

$$\text{trace}(AB^T) = s_1[\cos\varphi + e_1^2(1-\cos\varphi)] + s_2[\cos\varphi + e_2^2(1-\cos\varphi)] = s_1 e_1^2 + s_2 e_2^2 + \cos\varphi[s_1(1-e_1^2) + s_2(1-e_2^2)]. \tag{25}$$

This is clearly maximized for $\cos\phi = 1$, which means that $W = I$. Thus the optimal attitude is given by

$$A_{opt} \equiv UV^T. \tag{26}$$

Equation (25) shows that the minimum of trace(AB^T) is unique unless $s_2 = 0$. The vanishing of s_2 is the sign in the optimal algorithm that the observations are not sufficient to determine the attitude. We shall see below that this is related to the parallelism of the reference frame or body frame vectors.

The singular value decomposition is rather expensive computationally, so we look for a simpler way to compute A_{opr}. We note that the classical adjoint, or adjugate, (the transposed matrix of cofactors) of B^T is given in terms of the SVD by[16]

$$\text{adj}\, B^T = U[\text{diag}(0,0,s_1 s_2)]V^T. \tag{27}$$

We also note that

$$B\,B^T B = U[\text{diag}(s_1^3, s_2^3, 0)]V^T. \tag{28}$$

These allow us to write

$$(\lambda^2 - s_1 s_2)B + \lambda \text{adj}\, B^T - BB^T B = \lambda s_1 s_2 U V^T = \lambda s_1 s_2 A_{opt}, \tag{29}$$

where

$$\lambda \equiv s_1 + s_2 = \text{trace}(AB^T). \tag{30}$$

We can compute the optimal attitude without actually performing the expensive SVD of B if we can find an alternative means of computing the quantities appearing in equation (29). Direct computation from equation (17) gives

$$\text{adj}\, B^T = a_1 a_2 (\mathbf{b}_1 \times \mathbf{b}_2)(\mathbf{r}_1 \times \mathbf{r}_2)^T = a_1 a_2 |\mathbf{b}_1 \times \mathbf{b}_2||\mathbf{r}_1 \times \mathbf{r}_2| \mathbf{b}_3 \mathbf{r}_3^T. \tag{31}$$

Then we see from equation (27) that

$$s_1 s_2 = \|\text{adj}\, B^T\|_F = a_1 a_2 |\mathbf{b}_1 \times \mathbf{b}_2||\mathbf{r}_1 \times \mathbf{r}_2|, \tag{32}$$

where $\|M\|_F$ denotes the Frobenius (or Euclidean, or Schur, or Hilbert-Schmidt) norm[15,16]

$$\|M\|_F \equiv [\text{trace}(MM^T)]^{1/2}. \tag{33}$$

We note from equation (32) that $s_2 = 0$ if either of the cross products vanishes, as was mentioned above. A little effort is required to show that

$$\lambda^2 = s_1^2 + s_2^2 + 2s_1 s_2 = \|B\|_F^2 + 2a_1 a_2 |\mathbf{b}_1 \times \mathbf{b}_2||\mathbf{r}_1 \times \mathbf{r}_2| = a_1^2 + a_2^2 + 2a_1 a_2 \left[(\mathbf{b}_1 \cdot \mathbf{b}_2)(\mathbf{r}_1 \cdot \mathbf{r}_2) + |\mathbf{b}_1 \times \mathbf{b}_2||\mathbf{r}_1 \times \mathbf{r}_2|\right]. \tag{34}$$

In the two-observation case, λ is just the positive square root of the quantity on the right side of equation (34); finding λ in the case of more than two observations requires solving a quartic equation. To complete the analytic derivation, we need to evaluate

$$B\,B^T B = \sum_{i,j,k=1}^{2} a_i a_j a_k \mathbf{b}_i (\mathbf{r}_i \cdot \mathbf{r}_j)(\mathbf{b}_j \cdot \mathbf{b}_k)\mathbf{r}_k^T \tag{35}$$

Combining all these intermediate results with much vector algebra gives the final equation for the optimal attitude estimate:

$$A_{opt} \equiv (a_1/\lambda)[\mathbf{b}_1\mathbf{r}_1^T + (\mathbf{b}_1 \times \mathbf{b}_3)(\mathbf{r}_1 \times \mathbf{r}_3)^T] + (a_2/\lambda)[\mathbf{b}_2\mathbf{r}_2^T + (\mathbf{b}_2 \times \mathbf{b}_3)(\mathbf{r}_2 \times \mathbf{r}_3)^T] + \mathbf{b}_3\mathbf{r}_3^T. \tag{36}$$

It is interesting to note that this expression has a unique limit as either a_1 or a_2 goes to zero, with λ equal to the non-zero weight in the limit. This is true even though Wahba's loss function of equation (15) does not have a unique minimum in either limit, since it effectively only includes a single observation. In fact, the limit of the optimal estimate is the TRIAD estimate A_{T1} as a_2 goes to zero, and A_{T2} as a_1 goes to zero. It is also true, but more difficult to see, that the optimal estimate is equal to A_{T3} for equal weights, $a_1 = a_2$.

The optimal estimate maps the two reference vectors as

$$A_{opt}\mathbf{r}_1 = (a_1/\lambda)A_{T1}\mathbf{r}_1 + (a_2/\lambda)A_{T2}\mathbf{r}_1 = (a_1/\lambda)\mathbf{b}_1 + (a_2/\lambda)\left\{(\mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{b}_1 + [\mathbf{b}_2 - (\mathbf{b}_1 \cdot \mathbf{b}_2)\mathbf{b}_1]|\mathbf{r}_1 \times \mathbf{r}_2|/|\mathbf{b}_1 \times \mathbf{b}_2|\right\} \tag{37}$$

and

$$A_{opt}\mathbf{r}_2 = (a_1/\lambda)A_{T1}\mathbf{r}_2 + (a_2/\lambda)A_{T2}\mathbf{r}_2 = (a_2/\lambda)\mathbf{b}_2 + (a_1/\lambda)\left\{(\mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{b}_2 + [\mathbf{b}_1 - (\mathbf{b}_1 \cdot \mathbf{b}_2)\mathbf{b}_2]|\mathbf{r}_1 \times \mathbf{r}_2|/|\mathbf{b}_1 \times \mathbf{b}_2|\right\}. \tag{38}$$

The main point to note about these equations is that the optimal attitude estimate maps both $\mathbf{r}_1$ and $\mathbf{r}_2$ into the plane spanned by $\mathbf{b}_1$ and $\mathbf{b}_2$. It's clear from the loss function of equation (15) that this has to be the case; any out-of-plane component would be non-optimal.

In the case that $\mathbf{b}_1 \cdot \mathbf{b}_2 = \mathbf{r}_1 \cdot \mathbf{r}_2$, equation (34) for λ simplifies to $\lambda = a_1 + a_2$, and the optimal estimate is

$$A_{opt} = (a_1 A_{T1} + a_2 A_{T2})/(a_1 + a_2). \tag{39}$$

Since A_{T1} and A_{T2} are equal in this case, we see that A_{opt} is equal to their common value, also.

Mortari has found an alternative representation of the closed-form solution to the two-observation Wahba problem that is equivalent to the solution found here[17].

OPTIMIZED TRIAD

Bar-Itzhack and Harman[6] have proposed using equation (37) even when $\mathbf{b}_1 \cdot \mathbf{b}_2 \neq \mathbf{r}_1 \cdot \mathbf{r}_2$. In general, this estimator is not optimal, nor is the resulting attitude estimate exactly orthogonal. In order to produce a more nearly orthogonal attitude matrix, they employ the first-order orthogonalization step

$$A_{OT} = \tfrac{1}{2}[(a_1 + a_2)^{-1}(a_1 A_{T1} + a_2 A_{T2}) + (a_1 + a_2)(a_1 A_{T1}^T + a_2 A_{T2}^T)^{-1}] \tag{40}$$

They call the resulting estimator "Optimized TRIAD." This estimate has the correct limits of A_{T1} and A_{T2} as a_1 or a_2 tends to zero, respectively, but is not the same as A_{T3} for equal weights. It avoids the computation of λ that is required for the optimal estimate, but requires the inverse of a 3×3 matrix.

There is an alternative way to orthogonalize the matrix computed by equation (37) when $\mathbf{b}_1 \cdot \mathbf{b}_2 \neq \mathbf{r}_1 \cdot \mathbf{r}_2$. This is to extract a quaternion from the attitude matrix and then normalize the resulting quaternion. It is well known that the attitude matrix computed from a normalized quaternion is guaranteed to be orthogonal[10,11,18]. The extraction of the quaternion requires a square root, but it is often desirable to compute a quaternion for data transmission or storage, because it stores complete attitude information in four components instead of the nine required for the attitude matrix.

DIRECT QUATERNION METHOD

All the methods considered so far compute the attitude matrix. If a quaternion is desired, it can be extracted from the attitude matrix. However, it would be desirable to avoid this indirect and somewhat costly procedure. Reynolds has proposed a very simple estimation algorithm that computes a quaternion directly[7,8].

We first present some background information on quaternions to establish our conventions. A more complete discussion can be found in reference 11. A quaternion q has a vector part $\mathbf{q}$ and a scalar part q_s, which we write as

$$q = [\mathbf{q}, q_s]. \tag{41}$$

This is similar to Reynolds's notation except that we use square brackets rather than parentheses. A unit quaternion (*i.e.*, a quaternion with $|\mathbf{q}|^2 + q_s^2 = 1$) can be used to represent an attitude matrix, which rotates a vector by

$$A(q)\mathbf{v} = (q_s^2 - |\mathbf{q}|^2)\mathbf{v} + 2(\mathbf{q} \cdot \mathbf{v})\mathbf{q} - 2q_s(\mathbf{q} \times \mathbf{v}). \tag{42}$$

We will follow Shuster's convention for quaternion products[11], writing

$$p \otimes q = [\mathbf{p}, p_s] \otimes [\mathbf{q}, q_s] = [q_s\mathbf{p} + p_s\mathbf{q} - \mathbf{p} \times \mathbf{q},\ p_s q_s - \mathbf{p} \cdot \mathbf{q}]. \tag{43}$$

This differs from the historical convention in the sign of the cross-product, and has the advantage that the order of quaternion multiplication is the same as the order of attitude matrix multiplication:

$$A(p \otimes q) = A(p)A(q). \tag{44}$$

The quaternion corresponding to the rotation matrix $R(\mathbf{e},\phi)$ is

$$q = \left[\mathbf{e}\sin\frac{\phi}{2}, \cos\frac{\phi}{2}\right]. \tag{45}$$

The derivation of the direct quaternion method begins with the observation that the quaternion that maps the reference vector $\mathbf{r}_1$ into the body frame vector $\mathbf{b}_1$, using the minimum-angle rotation, is

$$q_{\min 1} \equiv \frac{1}{\sqrt{2(1+\mathbf{b}_1\cdot\mathbf{r}_1)}}[\mathbf{b}_1\times\mathbf{r}_1, 1+\mathbf{b}_1\cdot\mathbf{r}_1] \tag{46}$$

The most general matrix that maps $\mathbf{r}_1$ into $\mathbf{b}_1$ is $R(\mathbf{b}_1,\phi_b)A(q_{\min 1})R(\mathbf{r}_1,\phi_r)$, where ϕ_b and ϕ_r are arbitrary angles of rotation about $\mathbf{b}_1$ and $\mathbf{r}_1$, respectively. This general rotation has the quaternion representation

$$\begin{aligned} q_1 &= \frac{1}{\sqrt{2(1+\mathbf{b}_1\cdot\mathbf{r}_1)}}\left[\mathbf{b}_1\sin\frac{\phi_b}{2},\cos\frac{\phi_b}{2}\right]\otimes[\mathbf{b}_1\times\mathbf{r}_1, 1+\mathbf{b}_1\cdot\mathbf{r}_1]\otimes\left[\mathbf{r}_1\sin\frac{\phi_r}{2},\cos\frac{\phi_r}{2}\right] \\ &= \frac{1}{\sqrt{2(1+\mathbf{b}_1\cdot\mathbf{r}_1)}}\left[\cos\frac{\phi}{2}(\mathbf{b}_1\times\mathbf{r}_1)+\sin\frac{\phi}{2}(\mathbf{b}_1+\mathbf{r}_1),\ (1+\mathbf{b}_1\cdot\mathbf{r}_1)\cos\frac{\phi}{2}\right], \end{aligned} \tag{47}$$

where $\phi \equiv \phi_b + \phi_r$. By parallel reasoning, the most general quaternion that maps $\mathbf{r}_2$ into $\mathbf{b}_2$ is given by

$$q_2 = \frac{1}{\sqrt{2(1+\mathbf{b}_2\cdot\mathbf{r}_2)}}\left[\cos\frac{\psi}{2}(\mathbf{b}_2\times\mathbf{r}_2)+\sin\frac{\psi}{2}(\mathbf{b}_2+\mathbf{r}_2),\ (1+\mathbf{b}_2\cdot\mathbf{r}_2)\cos\frac{\psi}{2}\right] \tag{48}$$

for some angle ψ. The vector part of q_1 is perpendicular to $\mathbf{b}_1 - \mathbf{r}_1$, and the vector part of q_2 is perpendicular to $\mathbf{b}_2 - \mathbf{r}_2$. Based on this observation, Reynolds proposed to look for a quaternion whose vector part is perpendicular to both $\mathbf{b}_1 - \mathbf{r}_1$ and $\mathbf{b}_2 - \mathbf{r}_2$. The vector part of q_1 will be perpendicular to $\mathbf{b}_2 - \mathbf{r}_2$ if we choose

$$\cos(\phi/2) = \pm\{[(\mathbf{b}_1\times\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2)]^2+[(\mathbf{b}_1+\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2)]^2\}^{-1/2}(\mathbf{b}_1+\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2) \tag{49a}$$

and

$$\sin(\phi/2) = \mp\{[(\mathbf{b}_1\times\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2)]^2+[(\mathbf{b}_1+\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2)]^2\}^{-1/2}(\mathbf{b}_1\times\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2). \tag{49b}$$

Substituting this into equation (47) gives

$$q_1 = c_1^{-1/2}[(\mathbf{b}_1-\mathbf{r}_1)\times(\mathbf{b}_2-\mathbf{r}_2),\ (\mathbf{b}_1+\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2)], \tag{50}$$

where c_1 is the normalization factor

$$c_1 = |(\mathbf{b}_1-\mathbf{r}_1)\times(\mathbf{b}_2-\mathbf{r}_2)|^2+[(\mathbf{b}_1+\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2)]^2. \tag{51}$$

We have ignored the ambiguous overall sign of the quaternion, which has no significance, since the attitude matrix is a homogeneous quadratic function of the quaternion. The appearance of the cross product $(\mathbf{b}_1-\mathbf{r}_1)\times(\mathbf{b}_2-\mathbf{r}_2)$ is not at all surprising, since this vector is guaranteed to be orthogonal to both $\mathbf{b}_1 - \mathbf{r}_1$ and $\mathbf{b}_2 - \mathbf{r}_2$.

Similarly, choosing ψ so that the vector part of q_2 will be perpendicular to $\mathbf{b}_1 - \mathbf{r}_1$ gives

$$q_2 = c_2^{-1/2}[(\mathbf{b}_1-\mathbf{r}_1)\times(\mathbf{b}_2-\mathbf{r}_2),\ (\mathbf{b}_2+\mathbf{r}_2)\cdot(\mathbf{r}_1-\mathbf{b}_1)], \tag{52}$$

The vector parts of q_1 and q_2 are equal up to the normalization constant. However, the scalar part of q_1 is

$$q_{1s} = c_1^{-1/2}(\mathbf{b}_1+\mathbf{r}_1)\cdot(\mathbf{b}_2-\mathbf{r}_2) = c_1^{-1/2}[(\mathbf{b}_2\cdot\mathbf{r}_1-\mathbf{b}_1\cdot\mathbf{r}_2)+(\mathbf{b}_1\cdot\mathbf{b}_2-\mathbf{r}_1\cdot\mathbf{r}_2)] \tag{53}$$

and the scalar part of q_2 is

$$q_{2s} = c_2^{-1/2}(\mathbf{b}_2+\mathbf{r}_2)\cdot(\mathbf{r}_1-\mathbf{b}_1) = c_2^{-1/2}[(\mathbf{b}_2\cdot\mathbf{r}_1-\mathbf{b}_1\cdot\mathbf{r}_2)-(\mathbf{b}_1\cdot\mathbf{b}_2-\mathbf{r}_1\cdot\mathbf{r}_2)]. \tag{54}$$

Thus, q_1 and q_2 are identical if $\mathbf{b}_1\cdot\mathbf{b}_2 = \mathbf{r}_1\cdot\mathbf{r}_2$, and they are equal to

$$q_3 \equiv c_3^{-1/2}[(\mathbf{b}_1-\mathbf{r}_1)\times(\mathbf{b}_2-\mathbf{r}_2),\ \mathbf{b}_2\cdot\mathbf{r}_1-\mathbf{b}_1\cdot\mathbf{r}_2]. \tag{55}$$

We see that q_1, q_2, and q_3 all have the same rotation axis, and the rotation angle of q_3 is intermediate between those of q_1 and q_2. The quaternion q_3, which treats the two measurements symmetrically, is the estimate preferred by Reynolds; but we will also consider the asymmetrical estimates q_1 and q_2.

COMPARISON OF THE DIRECT QUATERNION METHOD WITH TRIAD

It would seem that the quaternion q_1 should correspond to the TRIAD estimate A_{T1}, q_2 to A_{T2}, and q_3 to A_{T3}. As evidence for this, we note that the direct quaternion estimation methods have $A(q_1)\mathbf{r}_1 = A_{T1}\mathbf{r}_1 = \mathbf{b}_1$, $A(q_2)\mathbf{r}_2 = A_{T2}\mathbf{r}_2 = \mathbf{b}_2$, and q_3 symmetric in the measurements, as A_{T3} is. However, we shall now show that this correspondence is not exact. The algebra in the general case becomes rather messy, so we consider a simple example. Assume that we have two reference vectors

$$\mathbf{r}_1 = [1,0,0]^T \quad \text{and} \quad \mathbf{r}_2 = [0,1,0]^T \tag{56}$$

and two observation vectors

$$\mathbf{b}_1 = [0,0,1]^T \quad \text{and} \quad \mathbf{b}_2 = [\cos\vartheta, 0, \sin\vartheta]^T . \tag{57}$$

We note that $\mathbf{b}_1 \cdot \mathbf{b}_2 = \mathbf{r}_1 \cdot \mathbf{r}_2$ only if $\sin\vartheta = 0$, in which case all algorithms should give the same estimate.

We first compute the TRIAD estimates. Straightforward algebra results in

$$A_{T1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_{T2} = \begin{bmatrix} -\sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \\ \cos\vartheta & \sin\vartheta & 0 \end{bmatrix}, \quad \text{and} \quad A_{T3} = \begin{bmatrix} -\sin(\vartheta/2) & \cos(\vartheta/2) & 0 \\ 0 & 0 & 1 \\ \cos(\vartheta/2) & \sin(\vartheta/2) & 0 \end{bmatrix} . \tag{58}$$

We note that $A_{T1}\mathbf{r}_1 = \mathbf{b}_1$, $A_{T2}\mathbf{r}_2 = \mathbf{b}_2$, $A_{T3}\mathbf{r}_+ = \mathbf{b}_+$, and $A_{T3}\mathbf{r}_- = \mathbf{b}_-$, as expected. However,

$$|A_{T1}\mathbf{r}_2 - \mathbf{b}_2| = |A_{T2}\mathbf{r}_1 - \mathbf{b}_1| = 2|\sin(\vartheta/2)|, \tag{59a}$$

and

$$|A_{T3}\mathbf{r}_1 - \mathbf{b}_1| = |A_{T3}\mathbf{r}_2 - \mathbf{b}_2| = 2|\sin(\vartheta/4)| . \tag{59b}$$

These results are not surprising, since the vectors $A_{T1}\mathbf{r}_2$, $A_{T2}\mathbf{r}_1$, $A_{T3}\mathbf{r}_1$, and $A_{T3}\mathbf{r}_2$ are all in the plane spanned by $\mathbf{b}_1$ and $\mathbf{b}_2$, as we argued was necessary for an optimal estimator. For comparison with the direct quaternion method, it is interesting to present the quaternions extracted from A_{T1}, A_{T2}, A_{T3}:

$$q_{T1} = \tfrac{1}{2}[1,1,1,1], \tag{60a}$$

$$q_{T2} = \tfrac{1}{2}\left[\sqrt{1-\sin\vartheta}, \sqrt{1+\sin\vartheta}, \sqrt{1+\sin\vartheta}, \sqrt{1-\sin\vartheta}\right], \tag{60b}$$

and

$$q_{T3} = \tfrac{1}{2}\left[\sqrt{1-\sin(\vartheta/2)}, \sqrt{1+\sin(\vartheta/2)}, \sqrt{1+\sin(\vartheta/2)}, \sqrt{1-\sin(\vartheta/2)}\right], \tag{60c}$$

where we have written out the three components of the vector part of each quaternion explicitly.

The estimates produced by the direct quaternion method embodied in equations (50), (52), and (55) are

$$q_1 = \tfrac{1}{2}(1+\cos\vartheta\sin\vartheta)^{-1/2}[1, \cos\vartheta + \sin\vartheta, 1, \cos\vartheta + \sin\vartheta], \tag{61a}$$

$$q_2 = \tfrac{1}{2}[1, \cos\vartheta + \sin\vartheta, 1, \cos\vartheta - \sin\vartheta], \tag{61b}$$

and

$$q_3 = (4 + 2\cos\vartheta\sin\vartheta - \sin^2\vartheta)^{-1/2}[1, \cos\vartheta + \sin\vartheta, 1, \cos\vartheta] . \tag{61c}$$

It is immediately apparent that the quaternions in equation (61) do not correspond to those in equation (60), unless $\sin\vartheta = 0$ and all reasonable estimators agree. The attitude matrices computed from q_1, q_2, and q_3 lead to further insights:

$$A(q_1) = \frac{1}{1+\cos\vartheta\sin\vartheta}\begin{bmatrix} 0 & \cos\vartheta + \sin\vartheta & -\cos\vartheta\sin\vartheta \\ 0 & \cos\vartheta\sin\vartheta & \cos\vartheta + \sin\vartheta \\ 1+\cos\vartheta\sin\vartheta & 0 & 0 \end{bmatrix}, \tag{62a}$$

$$A(q_2) = \begin{bmatrix} -\cos\vartheta\sin\vartheta & \cos\vartheta & \sin^2\vartheta \\ \sin\vartheta & 0 & \cos\vartheta \\ \cos^2\vartheta & \sin\vartheta & -\cos\vartheta\sin\vartheta \end{bmatrix}, \tag{62b}$$

and

$$A(q_3) = \frac{1}{4 + 2\cos\vartheta\sin\vartheta - \sin^2\vartheta}\begin{bmatrix} -\sin\vartheta(2\cos\vartheta + \sin\vartheta) & 2(2\cos\vartheta + \sin\vartheta) & -2\sin\vartheta(\cos\vartheta - \sin\vartheta) \\ 2\sin\vartheta & \sin\vartheta(2\cos\vartheta - \sin\vartheta) & 2(2\cos\vartheta + \sin\vartheta) \\ 4 + 2\sin\vartheta(\cos\vartheta - \sin\vartheta) & 2\sin\vartheta & -\sin\vartheta(2\cos\vartheta + \sin\vartheta) \end{bmatrix}. \quad (62c)$$

We note that $A(q_1)\mathbf{r}_1 = \mathbf{b}_1$ and $A(q_2)\mathbf{r}_2 = \mathbf{b}_2$, as expected. However, in the general case,

$$\sqrt{1 + \cos\vartheta\sin\vartheta}\,|A(q_1)\mathbf{r}_2 - \mathbf{b}_2| = |A(q_2)\mathbf{r}_1 - \mathbf{b}_1| = \sqrt{2}\,|\sin\vartheta| \quad (63a)$$

and

$$|A(q_3)\mathbf{r}_1 - \mathbf{b}_1| = |A(q_3)\mathbf{r}_2 - \mathbf{b}_2| = \sqrt{2}(4 + 2\cos\vartheta\sin\vartheta - \sin^2\vartheta)^{-1/2}|\sin\vartheta|. \quad (63b)$$

These residuals are all larger than the corresponding residuals in equation (59), because the vectors $A(q_1)\mathbf{r}_2$, $A(q_2)\mathbf{r}_1$, $A(q_3)\mathbf{r}_1$, and $A(q_3)\mathbf{r}_2$ all have components along the y axis in the body frame, which is perpendicular to the plane spanned by $\mathbf{b}_1$ and $\mathbf{b}_2$. According to our previous argument, they can't correspond to optimal estimates for any choice of weights. We may be prepared to give up optimality for computational simplicity, however.

SINGULARITY OF THE DIRECT QUATERNION METHOD

The direct quaternion method has the disadvantage of being ill determined whenever both the vector part and the scalar part of the estimated quaternion take the indeterminate value 0/0. We can easily see from equation (50) that q_1 is undefined if $\mathbf{b}_2 = \mathbf{r}_2$, which is when the axis of the attitude rotation is along $\mathbf{r}_2$ (and therefore is also along $\mathbf{b}_2$). Similarly, equation (52) shows that q_2 is undefined if $\mathbf{b}_1 = \mathbf{r}_1$, which is when the axis of the attitude rotation is along $\mathbf{r}_1$ and $\mathbf{b}_1$. These estimators are identical in the absence of measurement noise, and we certainly don't want to depend on measurement noise to avoid a singular condition. Thus we see that the direct quaternion method is singular whenever the attitude rotation axis is along $\mathbf{r}_1$ or $\mathbf{r}_2$ (or along $\mathbf{b}_1$ or $\mathbf{b}_2$). We will now show that the direct quaternion method is singular whenever the attitude rotation axis is in the $\mathbf{r}_1$, $\mathbf{r}_2$ plane, which means that it is also in the $\mathbf{b}_1$, $\mathbf{b}_2$ plane.

If neither $\mathbf{b}_1 - \mathbf{r}_1$ nor $\mathbf{b}_2 - \mathbf{r}_2$ is zero, the vector part of the quaternion estimate vanishes if they are parallel, that is, if

$$\mathbf{b}_2 - \mathbf{r}_2 = \beta(\mathbf{b}_1 - \mathbf{r}_1) \quad (64)$$

for some scalar β. The vector $\mathbf{b}_2 = \mathbf{r}_2 + \beta(\mathbf{b}_1 - \mathbf{r}_1)$ has unit norm, which means that

$$1 = 1 + 2\beta\mathbf{r}_2 \cdot (\mathbf{b}_1 - \mathbf{r}_1) + 2\beta^2(1 - \mathbf{b}_1 \cdot \mathbf{r}_1). \quad (65)$$

Solving this for β (the zero root is not allowed since $\mathbf{b}_2 - \mathbf{r}_2 \neq 0$) and substituting into equation (64) gives

$$\mathbf{b}_2 = \mathbf{r}_2 - [\mathbf{r}_2 \cdot (\mathbf{b}_1 - \mathbf{r}_1)/(1 - \mathbf{b}_1 \cdot \mathbf{r}_1)](\mathbf{b}_1 - \mathbf{r}_1). \quad (66)$$

It is now straightforward to show that equation (2) is obeyed and that

$$\mathbf{b}_2 \cdot \mathbf{r}_1 = \mathbf{b}_1 \cdot \mathbf{r}_2. \quad (67)$$

Thus the vanishing of the vector part of the quaternion estimates of equations (50), (52), and (55) ensures that the scalar parts vanish automatically.

Now let us see what these conditions imply about the attitude quaternion, which certainly exists even if it cannot be computed by the direct quaternion method. Equation (42) requires

$$\mathbf{b}_i = (q_s^2 - |\mathbf{q}|^2)\mathbf{r}_i + 2(\mathbf{q} \cdot \mathbf{r}_i)\mathbf{q} - 2q_s(\mathbf{q} \times \mathbf{r}_i) \quad \text{for } i = 1, 2. \quad (68)$$

From this equation, we can see that

$$\mathbf{b}_2 \cdot \mathbf{r}_1 - \mathbf{b}_1 \cdot \mathbf{r}_2 = 4q_s\mathbf{q} \cdot (\mathbf{r}_1 \times \mathbf{r}_2). \quad (69)$$

This means that the scalar part of the direct quaternion estimate vanishes either if $\mathbf{q}$ is perpendicular to $\mathbf{r}_1 \times \mathbf{r}_2$, which is to say that it is in the $\mathbf{r}_1$, $\mathbf{r}_2$ plane, or else if q_s is zero, which indicates a 180° rotation. We still have to investigate the requirement that $\mathbf{b}_1 - \mathbf{r}_1$ is parallel to $\mathbf{b}_2 - \mathbf{r}_2$. If $\mathbf{q}$ is in the $\mathbf{r}_1$, $\mathbf{r}_2$ plane, we can write

$$\mathbf{q} = \alpha_1\mathbf{r}_1 + \alpha_2\mathbf{r}_2. \quad (70)$$

With equation (68), this gives

$$\mathbf{b}_1 - \mathbf{r}_1 = 2\alpha_2[-(\alpha_2 + \alpha_1 \mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{r}_1 + (\alpha_1 + \alpha_2 \mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{r}_2 + q_s(\mathbf{r}_1 \times \mathbf{r}_2)] \tag{71a}$$

and

$$\mathbf{b}_2 - \mathbf{r}_2 = -2\alpha_1[-(\alpha_2 + \alpha_1 \mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{r}_1 + (\alpha_1 + \alpha_2 \mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{r}_2 + q_s(\mathbf{r}_1 \times \mathbf{r}_2)]. \tag{71b}$$

These two vectors are clearly parallel. On the other hand, equation (68) for a 180° rotation gives

$$\mathbf{b}_i - \mathbf{r}_i = -2\mathbf{r}_i + 2(\mathbf{q} \cdot \mathbf{r}_i)\mathbf{q} = 2\mathbf{q} \times (\mathbf{q} \times \mathbf{r}_i), \tag{72}$$

and a straightforward but tedious calculation gives

$$(\mathbf{b}_1 - \mathbf{r}_1) \times (\mathbf{b}_2 - \mathbf{r}_2) = 4[\mathbf{q} \cdot (\mathbf{r}_1 \times \mathbf{r}_2)]\mathbf{q}. \tag{73}$$

Thus the attitude rotation axis is required to be in the $\mathbf{r}_1$, $\mathbf{r}_2$ plane for the 180° rotation case to be singular, also. Thus we have completely characterized the singular cases of the direct quaternion method as those cases for which the attitude rotation axis is in the $\mathbf{r}_1$, $\mathbf{r}_2$ plane, and therefore in the $\mathbf{b}_1$, $\mathbf{b}_2$ plane, also.

The direct quaternion estimate method is singular if the attitude matrix is the identity, giving $\mathbf{r}_1 = \mathbf{b}_1$ and $\mathbf{r}_2 = \mathbf{b}_2$. We can say that the rotation axis is in the $\mathbf{r}_1$, $\mathbf{r}_2$ plane in this case, also, because the rotation axis can be arbitrarily assigned for zero rotation angle. Reynolds has proposed a method to avoid the singular condition in most cases, but it does not avoid the singularity for attitude matrices close to the identity[7,8]. The singular condition can be avoided in all cases by applying Shuster's method of sequential rotations[10,19]. This method solves for the attitude with respect to reference coordinate frames rotated from the original frame by 180° about the *x*, *y*, or *z* coordinate axis. That is, we solve for the quaternions

$$q^i \equiv q \otimes [\mathbf{e}_i, 0] = [\mathbf{q}, q_s] \otimes [\mathbf{e}_i, 0] = [q_s \mathbf{e}_i - \mathbf{q} \times \mathbf{e}_i, -\mathbf{q} \cdot \mathbf{e}_i] \quad \text{for } i = 1, 2, 3, \tag{74}$$

where $\mathbf{e}_i$ is the unit vector along the i^{th} coordinate axis. These rotations are easy to implement on the reference vectors, since they simply change the signs of the components perpendicular to $\mathbf{e}_i$. Merely permuting and changing signs of the components of the rotated quaternion recovers the unrotated quaternion. For example

$$q^1 = [q_1, q_2, q_3, q_s] \otimes [1, 0, 0, 0] = [q_s, -q_3, q_2, -q_1]. \tag{75}$$

The method of sequential rotations always avoids the singularity, since the 3×4 matrix

$$[\mathbf{q} \,\vdots\, q_s \mathbf{e}_1 - \mathbf{q} \times \mathbf{e}_1 \,\vdots\, q_s \mathbf{e}_2 - \mathbf{q} \times \mathbf{e}_2 \,\vdots\, q_s \mathbf{e}_3 - \mathbf{q} \times \mathbf{e}_3] \tag{76}$$

always has rank three, as can be seen with some effort. Thus the rotation axes produced by the method of sequential rotations span the entire three-dimensional space, which means that they cannot all be coplanar with $\mathbf{r}_1$ and $\mathbf{r}_2$.

To use Shuster's rotations to avoid the singularity, we compute the reference vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ in all four reference frames, unrotated and rotated about the *x*, *y*, and *z* axes. We compute the magnitude squared of the cross product $(\mathbf{b}_1 - \mathbf{r}_1) \times (\mathbf{b}_2 - \mathbf{r}_2)$ in each frame, and evaluate the quaternion in the frame where the cross product has the largest magnitude. The above analysis shows that this should provide the most robust estimate. If the optimal reference frame is not the unrotated frame, we recover the desired quaternion that transforms the unrotated reference frame to the spacecraft body frame by using equation (75) or its equivalent for other rotations.

COMPUTATIONAL EFFORT

The speed comparison is based on the floating point operation (flop) counts in MATLAB implementations of the algorithms, which have the advantage of being platform-independent. There are some caveats to make with regard to timing comparisons. First, for ground computations, absolute speed isn't all that important, since the estimation algorithm is only a part of the overall attitude determination data processing effort. Speed was more important in the past, when thousands of attitude solutions had to be computed by slower machines. Second, for real-time processing, as for an attitude control system onboard a spacecraft, the longest time is more important than the average time, because the attitude control system processor has to finish its task in a limited amount of time. This works against methods that may require sequential rotations.

Four methods for computing the attitude matrix are compared in Table 1: asymmetric TRIAD of equation (8), symmetric TRIAD of equation (12), the optimal two-measurement estimator of equation (36), and Optimized TRIAD of equation (40). An "asymmetric" estimator maps one of the two reference vectors into the corresponding observed vector exactly, throwing all the measurement errors into the other vector. A "symmetric" estimator, on the other hand, treats the two measurements symmetrically. The cost of using these four estimators to produce a quaternion is also presented. Every algorithm except

Optimized TRIAD computes the quaternion by extracting it from the corresponding attitude matrix, a process that costs 29 flops (see the Appendix). The quaternion output of Optimized TRIAD is cheaper than the attitude matrix output because it is extracted from the estimate of equation (39) rather than from equation (40). In addition to these four estimators, three other estimators are included for quaternion output only: the asymmetric direct quaternion estimator of equation (50), the symmetric direct quaternion estimator of equation (55), and QUEST, for comparison[12]. The computational effort for the direct quaternion estimation methods is given both with and without the use of rotations to avoid singular configurations. The computational effort of QUEST does not include the cost of sequential rotations. No special efforts have been made to achieve the most efficient possible implementation of any of the algorithms.

Table 1: Computational Effort of Estimation Algorithms in Flops

Algorithm	*A* output	*q* output
Asymmetric TRIAD	143	172
Symmetric TRIAD	166	195
Optimal Two-Measurement Estimator	265	294
Optimized TRIAD	335	273
Asymmetric Direct Quaternion	——	46
Asymmetric Direct Quaternion with Singularity Avoidance	——	108
Symmetric Direct Quaternion	——	50
Symmetric Direct Quaternion with Singularity Avoidance	——	112
QUEST	——	190

Several conclusions are apparent from these results. Symmetric estimators are a little more expensive than asymmetric estimators, in general. Optimized TRIAD with the approximate matrix orthogonalization of equation (40) is significantly more expensive than the optimal two-measurement estimator. If quaternion output is desired, Optimized TRIAD is slightly less expensive than the optimal two-measurement estimator; but the savings are less than 10%. However, the optimal two-measurement estimator and Optimized TRIAD (and even symmetric TRIAD) require more computational effort than QUEST to produce a quaternion. Asymmetric TRIAD is only slightly less expensive than QUEST, but the direct quaternion estimation methods developed by Reynolds are significantly faster. The implementation of rotations to avoid singularities in the direct quaternion estimation methods more than doubles their computational cost, but they are faster than other methods even with this modification. None of the three algorithms faster than QUEST is optimal, though; and QUEST also has the advantage of being a general-purpose algorithm applicable to any number of measurements, which avoids the need to develop and test a special-purpose two-observation algorithm.

ACCURACY

We will analyze two test scenarios, using the nine estimators with quaternion output that were used in the timing tests. The first scenario simulates two star trackers with narrow fields of view and orthogonal boresights at $[1, 0, 0]^T$ and $[0, 1, 0]^T$. We shall assume that the first tracker is tracking five stars at

$$\mathbf{b}_{11} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_{12} = \begin{bmatrix} 0.99712 \\ 0.07584 \\ 0 \end{bmatrix}, \quad \mathbf{b}_{13} = \begin{bmatrix} 0.99712 \\ -0.07584 \\ 0 \end{bmatrix}, \quad \mathbf{b}_{14} = \begin{bmatrix} 0.99712 \\ 0 \\ 0.07584 \end{bmatrix}, \quad \text{and} \quad \mathbf{b}_{15} = \begin{bmatrix} 0.99712 \\ 0 \\ -0.07584 \end{bmatrix}, \tag{77a}$$

and the second tracker is tracking three stars at

$$\mathbf{b}_{21} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{b}_{22} = \begin{bmatrix} 0 \\ 0.99712 \\ 0.07584 \end{bmatrix}, \quad \text{and} \quad \mathbf{b}_{23} = \begin{bmatrix} 0 \\ 0.99712 \\ -0.07584 \end{bmatrix}, \tag{77b}$$

We simulate 1000 test cases with random attitude matrices. We use the attitude matrices to map the eight observation vectors to the reference frame, add Gaussian random noise with equal standard deviations of 6 arcseconds per axis to the reference vectors, and then normalize them. The errors are unconventionally applied to the reference vectors rather than the observation vectors so that equation (77) will remain valid in the presence of noise. The two-observation estimators use averages of the multiple vectors observed by each tracker, as suggested by Bronzenac and Bender[5]. In this example the two averaged vectors in the body frame are along the star tracker boresights. The optimal estimator weights for these estimators are proportional to the inverse measurement variances, or to the number of vectors included in the average, so we use $a_2 = 0.6\,a_1$ for the optimal two-measurement estimator and Optimized TRIAD.

We treat the eight star measurements independently in QUEST, rather than averaging them. QUEST requires 316 flops for eight measurements, but avoids the cost of averaging the vectors, which is 108 flops. Thus the computational effort of QUEST should be taken as 208 flops for comparison with the other estimators in this eight-measurement example, making it more expensive than the direct quaternion estimator and TRIAD, but faster than the optimal two-vector estimator and Optimized TRIAD. In these tests, QUEST always used information about the true quaternion to determine the optimally rotated reference frame for estimation, without the need to perform sequential rotations.

Table 2 shows that symmetric TRIAD, the optimal two-measurement estimator, and Optimized TRIAD perform almost as well as QUEST. This justifies Bronzenac and Bender's procedure of using average observation and measurement vectors for the two star trackers. It should be noted, however, that the choice of orthogonal tracker boresights is optimal for this approximation, and that symmetric TRIAD is the only one of these algorithms that is computationally less expensive than QUEST, requiring 13 fewer flops. The symmetric direct quaternion estimator with singularity avoidance provides average and maximum errors within 10% of those of the best estimators with less computational effort, though.

Table 2: Average (Maximum) Estimation Errors (arcseconds) for Star Tracker Scenario

Algorithm	All Cases	$\lvert q_3 \rvert \geq 1/2$	$\lvert q_3 \rvert < 1/2$
Asymmetric TRIAD	4.6 (12.1)	4.5 (11.3)	4.7 (12.1)
Asymmetric Direct Quaternion	13.6 (2562)	5.2 (17.8)	20.1 (2562)
Asymmetric Direct Quaternion with Singularity Avoidance	5.1 (16.9)	5.1 (14.5)	5.1 (16.9)
Symmetric TRIAD	4.4 (12.2)	4.3 (11.6)	4.5 (12.2)
Symmetric Direct Quaternion	14.2 (4763)	4.7 (14.6)	21.6 (4763)
Symmetric Direct Quaternion with Singularity Avoidance	4.7 (12.9)	4.6 (12.1)	4.8 (12.9)
Optimized TRIAD	4.6 (12.1)	4.5 (11.3)	4.7 (12.1)
Optimal Two-Measurement Estimator	4.6 (12.1)	4.5 (11.3)	4.7 (12.1)
QUEST	4.4 (11.8)	4.3 (11.5)	4.4 (11.8)

The results also show that symmetric estimators perform slightly better than asymmetric estimators in this scenario. This was expected, since the number of stars tracked in the two trackers and thus the measurement weights are nearly equal. A symmetric estimator would be preferred in a real star tracker application, since there would be no way of predicting *a priori* which tracker would view more stars.

Table 2 also shows inferior performance of the direct quaternion estimators without singularity avoidance. The performance is not so bad in the 436 simulated cases with $\lvert q_3 \rvert \geq 1/2$ as in the 564 cases with $\lvert q_3 \rvert < 1/2$. The latter are the cases in which we would expect singularities to occur, since they have either small rotation angles or rotation axes close to the *x-y* plane, the $\mathbf{b}_1$, $\mathbf{b}_2$ plane in this scenario. This shows the importance of avoiding singular cases in an application of these estimators. We note that the performance with singularity avoidance, as well as the performance of all the other estimators, is independent of q_3.

The second scenario that we consider is a sun-mag system, similar to that on SAMPEX[3], assuming a digital sun sensor with accuracy of 0.1° and a magnetometer with effective accuracy of 1°. We assume that the Sun is at the center of view of the digital sun sensor at $\mathbf{b}_1 = [1, 0, 0]^T$, but the orientation of the magnetic field vector is not fixed in the spacecraft body frame. We simulate 1000 random attitude matrices and random magnetic field vector orientations, except that we do not allow the magnetic field direction to be within 5° of the ± y axis. These are the cases with coaligned vectors that the SAMPEX onboard attitude determination system rejects. We use the attitude matrices to map the Sun and magnetic field observation vectors to the reference frame, add Gaussian random noise with standard deviations as specified above, and then normalize the reference vectors. In this case the optimal estimator weights have $a_2 = 0.01\, a_1$.

The estimation errors for this scenario are presented in Table 3. The roll (x axis) and pitch/yaw (root-sum-squared of y and z axes) errors are presented separately, since the estimate of pitch and yaw provided by the digital sun sensor on the x axis is more precise than the roll angle estimate provided by the magnetometer. We note from these tables that QUEST and the optimal two-measurement estimator give identical errors, as they must since this scenario has two vector measurements. Since the weight assigned to the magnetometer measurement is so much less than the weight of the sun sensor measurement, Optimized TRIAD and asymmetric TRIAD give virtually the same results as the optimal estimators. The asymmetric direct quaternion estimator with singularity avoidance provides equivalent pitch and yaw errors, and average and maximum roll errors within 20% of those of the best estimators, with less computational effort.

Symmetric estimators are inferior to asymmetric estimators in the sun-mag scenario, since they allow the magnetometer errors to corrupt the sun sensor determination of pitch and yaw. Table 3a shows that the direct quaternion estimation method must be modified to provide acceptable roll estimation in the 551 cases with $|\mathbf{q}_\perp| < 1/2$, where $\mathbf{q}_\perp$ is the component of $\mathbf{q}$ perpendicular to the $\mathbf{b}_1$, $\mathbf{b}_2$ plane. Table 3b shows that pitch and yaw estimates provided by the asymmetric direct quaternion estimator are insensitive to these singular configurations, since this estimator maps $\mathbf{r}_1$ into $\mathbf{b}_1$ exactly.

Table 3a: Average (Maximum) Roll Estimation Errors (degrees) for Sun-Mag Test Case

Algorithm	All Cases	$\lvert\mathbf{q}_\perp\rvert \geq 1/2$	$\lvert\mathbf{q}_\perp\rvert < 1/2$
Asymmetric TRIAD	0.88 (3.06)	0.93 (2.92)	0.84 (3.06)
Asymmetric Direct Quaternion	2.82 (114)	1.07 (4.78)	4.24 (114)
Asymmetric Direct Quaternion with Singularity Avoidance	1.01 (3.58)	1.05 (3.36)	0.98 (3.58)
Symmetric TRIAD	0.88 (3.06)	0.93 (2.92)	0.84 (3.06)
Symmetric Direct Quaternion	1.97 (86.5)	0.98 (3.78)	2.77 (86.5)
Symmetric Direct Quaternion with Singularity Avoidance	0.92 (3.23)	0.96 (3.16)	0.89 (3.23)
Optimized TRIAD	0.88 (3.06)	0.93 (2.92)	0.84 (3.06)
Optimal Two-Measurement Estimator	0.88 (3.06)	0.93 (2.92)	0.84 (3.06)
QUEST	0.88 (3.06)	0.93 (2.92)	0.84 (3.06)

Table 3b: Average (Maximum) Pitch/Yaw Estimation Errors (degrees) for Sun-Mag Test Case

Algorithm	All Cases	$\lvert\mathbf{q}_\perp\rvert \geq 1/2$	$\lvert\mathbf{q}_\perp\rvert < 1/2$
Asymmetric TRIAD	0.13 (0.36)	0.13 (0.35)	0.13 (0.36)
Asymmetric Direct Quaternion	0.13 (0.36)	0.13 (0.35)	0.13 (0.36)
Asymmetric Direct Quaternion with Singularity Avoidance	0.13 (0.36)	0.13 (0.35)	0.13 (0.36)
Symmetric TRIAD	0.43 (1.60)	0.42 (1.54)	0.43 (1.60)
Symmetric Direct Quaternion	1.53 (96.3)	0.50 (1.91)	2.37 (96.3)
Symmetric Direct Quaternion with Singularity Avoidance	0.48 (1.92)	0.49 (1.91)	0.48 (1.92)
Optimized TRIAD	0.13 (0.37)	0.13 (0.35)	0.13 (0.37)
Optimal Two-Measurement Estimator	0.13 (0.37)	0.13 (0.35)	0.13 (0.37)
QUEST	0.13 (0.37)	0.13 (0.35)	0.13 (0.37)

CONCLUSIONS

We have analyzed four spacecraft attitude determination methods using exactly two vector measurements: the well-known TRIAD algorithm, an optimal closed-form two-measurement of Wahba's optimization problem, the Optimized TRIAD algorithm of Bar-Itzhack and Harman, and the direct quaternion estimation method of Reynolds. These methods are applicable to a variety of problems, including coarse "sun-mag" attitude estimation using the unit vector to the Sun and the Earth's magnetic field vector and precise estimation using unit vectors to stars tracked by two star trackers. For TRIAD and the direct quaternion estimation method, we investigate both "asymmetric" forms that map one of the two reference vectors into the corresponding observed vector exactly, and "symmetric" forms that distribute the errors symmetrically between the two measurements. We also include the well-known QUEST algorithm for comparison.

The computational speed of the algorithms was compared using floating point operation (flop) counts in MATLAB. These show that Optimized TRIAD and the optimal two-measurement estimator are more expensive than QUEST, which has the additional advantage of being a general-purpose algorithm applicable to any number of measurements. The direct quaternion estimation methods are significantly faster than QUEST, however. Both QUEST and the direct quaternion estimation methods have the disadvantage of sometimes requiring special computations to avoid singular cases, but the direct quaternion estimation methods are faster than any other methods even with these modifications.

We analyzed the accuracy of the estimators in two test scenarios. The first scenario simulated two star trackers with narrow fields of view and orthogonal boresights, using average vectors for five stars in the first tracker and three in the second. The second scenario simulated a digital sun sensor with accuracy of 0.1° and a magnetometer with effective accuracy of 1°. Symmetric estimators outperformed asymmetric estimators in the first scenario, and asymmetric estimators were superior in the second, as was expected. With this proviso, all the estimators had comparable errors. The one exception is that the direct quaternion estimators had larger errors if not modified to avoid singularities, showing the need for these modifications.

This paper demonstrates the superiority of TRIAD, QUEST, and the direct quaternion estimation methods for attitude estimation from two vector measurements.

REFERENCES

1. Black, Harold D., "A Passive System for Determining the Attitude of a Satellite," *AIAA Journal*, Vol. 2, No. 7, July 1964, pp. 1350–1351.
2. Lerner, Gerald M., "Three-Axis Attitude Determination," in *Spacecraft Attitude Determination and Control*, James R. Wertz, ed., D. Reidel, Dordrecht, Holland 1978.
3. Flatley, Thomas W., Josephine K. Forden, Debra A. Henretty, E. Glenn Lightsey, and F. Landis Markley, "Onboard Attitude Determination and Control Algorithms for SAMPEX," Flight Mechanics/ Estimation Theory Symposium, Goddard Space Flight Center, Greenbelt, MD, May 1990, NASA Conference Publication 3102.
4. Wahba, Grace, "A Least Squares Estimate of Spacecraft Attitude," *SIAM Review*, Vol. 7, No. 3, July 1965, p. 409.
5. Bronzenac, Thomas F. and Douglas J. Bender, "A Simple Suboptimal Least-Squares Algorithm for Attitude Determination with Multiple Sensors," Flight Mechanics/ Estimation Theory Symposium, Goddard Space Flight Center, Greenbelt, MD, May 1994, NASA Conference Publication 3265.
6. Bar-Itzhack, Itzhack Y. and Richard R. Harman, "Optimized TRIAD Algorithm for Attitude Determination," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, January-February 1997, pp. 208-211.
7. Reynolds, R. G., "Quaternion Parameterization and Global Attitude Estimation," Flight Mechanics Symposium, Goddard Space Flight Center, Greenbelt, MD, May 1997, NASA Conference Publication 3345.
8. Reynolds, R. G., "Quaternion Parameterization and a Simple Algorithm for Global Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, July-August 1998, pp. 669-671.
9. Keat, J., "Analysis of Least Squares Attitude Determination Routine DOAOP," Computer Sciences Corporation, CSC/TM-77/6034, February 1977.
10. Markley, F. Landis, "Parameterization of the Attitude," in *Spacecraft Attitude Determination and Control*, James R. Wertz, ed., D. Reidel, Dordrecht, Holland 1978.
11. Shuster, Malcolm D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4 October-December 1993, pp. 439-517.

12. Shuster, M. D. and S. D. Oh, "Three-Axis Attitude Determination from Vector Observations," *Journal of Guidance and Control*, Vol. 4, No. 1, January-February 1981, pp. 70-77.

13. Markley, F. Landis, "Attitude Determination Using Vector Observations: A Fast Optimal Matrix Algorithm," *Journal of the Astronautical Sciences*, Vol. 41, No. 2, April-June 1993, pp. 261-280.

14. Markley, F. Landis, "Attitude Determination Using Vector Observations and the Singular Value Decomposition," *Journal of the Astronautical Sciences*, Vol. 36, No. 3, July-September 1988, pp. 245-258.

15. Golub, Gene H. and Charles F. Van Loan, *Matrix Computations*, the Johns Hopkins University Press, Baltimore, 1983.

16. Horn, Roger A. and Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985

17. Mortari, Daniele, "*EULER*-2 and *EULER-n* Algorithms for Attitude Determination from Vector Observations," *Space Technology*, Vol. 16, No. 5/6, 1996, pp. 317-3121.

18. Giardina, C. R., R. Bronson, and L. Wallen, "An Optimal Normalization Scheme," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-11, No. 4, July 1975, pp. 443-446.

19. Shuster, Malcolm D. and Gregory A. Natanson, "Quaternion Computation from a Geometric Point of View," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, October-December 1993, pp. 545-556.

APPENDIX

The following MATLAB function was used to extract quaternions from attitude matrices[10,11].

```
function q = dcm2quat(a)

% finds the quaternion representation of a direction cosine matrix a

% find maximum of trace or diagonal element of direction cosine matrix
tra    = trace(a);
[mx,i] = max([a(1,1) a(2,2) a(3,3) tra]);

% compute unnormalized quaternion
if i==1, q = [2*mx+1-tra;a(1,2)+a(2,1);a(1,3)+a(3,1);a(2,3)-a(3,2)]; end;
if i==2, q = [a(2,1)+a(1,2);2*mx+1-tra;a(2,3)+a(3,2);a(3,1)-a(1,3)]; end;
if i==3, q = [a(3,1)+a(1,3);a(3,2)+a(2,3);2*mx+1-tra;a(1,2)-a(2,1)]; end;
if i==4, q = [a(2,3)-a(3,2);a(3,1)-a(1,3);a(1,2)-a(2,1);1+tra];      end;

% normalize the quaternion
q = q/norm(q);
```

AN ADAPTIVE KALMAN FILTER USING A SIMPLE RESIDUAL TUNING METHOD

Richard R. Harman
NASA-GSFC
Code 572
Aerospace Engineer
Greenbelt, MD 20771

One difficulty in using Kalman filters in real world situations is the selection of the correct process noise, measurement noise, and initial state estimate and covariance. These parameters are commonly referred to as tuning parameters. Multiple methods have been developed to estimate these parameters. Most of those methods such as maximum likelihood, subspace, and observer Kalman Identification require extensive offline processing and are not suitable for real time processing. One technique, which is suitable for real time processing, is the residual tuning method. Any mismodeling of the filter tuning parameters will result in a non-white sequence for the filter measurement residuals. The residual tuning technique uses this information to estimate corrections to those tuning parameters. The actual implementation results in a set of sequential equations that run in parallel with the Kalman filter. A. H. Jazwinski developed a specialized version of this technique for estimation of process noise ["Adaptive Filtering", Proceedings of IFAC Symposium Multivariable Control Systems, Dusseldorf, Germany, October 1968, Vol. 2, pp. 1-15]. Equations for the estimation of the measurement noise have also been developed. These algorithms are used to estimate the process noise and measurement noise for the Wide Field Infrared Explorer star tracker and gyro.

Vornoi Density Reduction Method to Select Stars for an On-board Catalog

Robert Bauer, *Bauer* Engineering Enterprises, Langhorne, Pennsylvania

Introduction

In Ref. 1 Bauer introduces the *Voronoi density reduction* method to select stars for an on-board catalog from a larger candidate set. He compares the method to previous methods of Vedder, Yates, and Kudva (Refs. 2, 3, 4) by way of three illustrative examples. Here a fourth example is given which compares the VDR method to a common approach in which a circular window is swept over the sky, and the 3 brightest stars are selected for the onboard catalog.

The Voronoi density reduction method is based on the *Voronoi diagram* of candidate stars. The Voronoi diagram is a subdivision of the celestial sphere into spherical polygonal cells, one for each star, so that the cell for star **P** consists of the region closer to **P** than to any other star. The star having the smallest *Voronoi cell* is considered to be in the densest region of the candidate set. The topological dual of the Voronoi diagram is the Delaunay triangulation. It has the important characteristic that the spherical cap circumscribing any triangle is devoid of other candidate stars. These *Delaunay caps* represent the "holes" in the candidate set.

The Voronoi density reduction method has been implemented as the Vorosel™ (vor-o-SEL) Tool. The user's manual for the tool is given in the Appendix. In the sequel it is assumed that the reader has read the introduction to the user's manual.

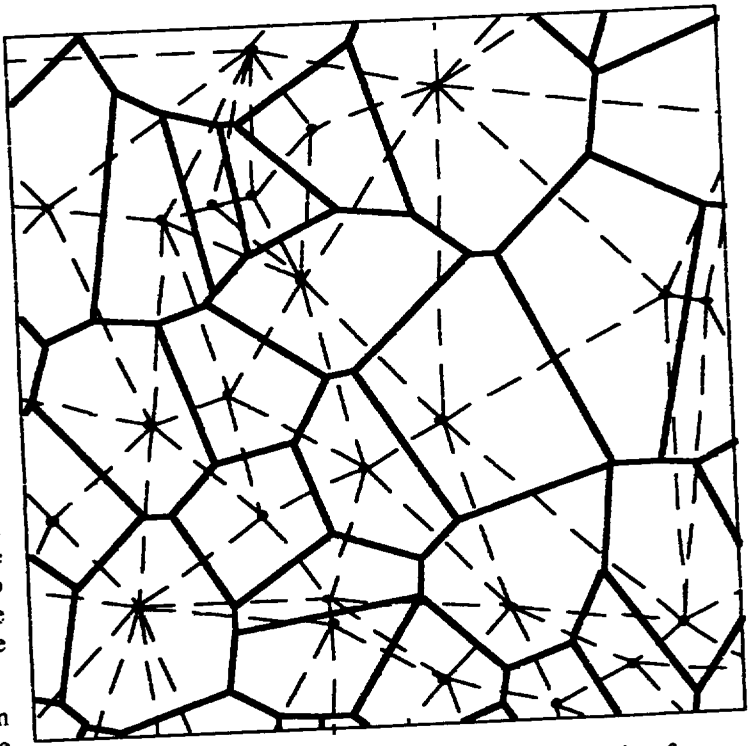


Figure 1 Voronoi Diagram/Delaunay Triangulation for a Set of Randomly Distributed Points.

An Example

For the example a candidate set of 5199 stars were artificially generated by randomly distributing stars over the unit sphere. The density of stars near the poles was made to be about 40% less than near the equator. The magnitudes of the stars were randomly distributed between 1 and 6 using a distribution typical of actual stars. An onboard catalog was then selected using the following method. A 5.0 degree radius circular star sensor field-of-view (FOV) was swept uniformly over the sphere. The three brightest stars in the FOV at any given position were selected to generate the "KHQS" (Keep High Quality Stars) catalog. This method will favor brighter stars and guarantee to the extent possible that the sensor sees at least three catalog stars for any position of its FOV. A total of 2592 stars were selected, an average of 4.9 stars per FOV.

The Vorosel Tool was then used to generate three other onboard catalogs with the same number of stars as the KHQS catalog. The "SVDR" catalog was generated using the standard Voronoi density reduction method without any options, i.e. no star weighting, Delaunay cap anchoring, etc. As will be demonstrated below, the SVDR catalog has a much more uniform spatial distribution of stars than the KHQS catalog, but also includes a higher frequency of dimmer stars.

The "WVDR" catalog was generated using the Voronoi density reduction method with star weighting applied to favor brighter stars. The weight for the i^{th} star was set according to the formula below:

$$w_i = exp(-g\, m_i) \qquad (1)$$

where m_i is the star magnitude and $g = 0.8$ is the weighting parameter. The WVDR catalog has approximately the same star magnitude distribution as the KHQS catalog, but has a more uniform spatial distribution.

Finally, the "DCA" catalog was generated using Voronoi density reduction with Delaunay cap anchoring. Before the Voronoi density reduction was applied, candidate stars defining Delaunay caps having radii greater than 3.33 degree were anchored. A total of 1026 stars were anchored in this manor. Then during the Voronoi density reduction, if the removal of a star from the candidate set created a Delaunay cap having radius greater than 3.33 degree the star was reinserted and anchored. In this manor 27 additional stars were anchored. The remaining 1539 stars were selected by the Voronoi density reduction. The 3.33 degree limit for Delaunay cap anchoring was chosen to provide a high frequency of at least three stars in the sensor FOV, explained as follows. If the spatial distribution of stars in the catalog were ideally uniform the stars would be located at the vertices of equilateral triangles. The smallest number of such stars needed to assure at least three stars in the sensor FOV would occur when the length of the sides of the equilateral triangles were equal to the sensor FOV radius. The radius of a cap circumscribing one of the equilateral triangle is two-thirds the sensor FOV radius or 3.33 degrees. Amoung the catalogs generated by the Vorosel tool, the DCA catalog has the highest frequency of three or more stars in the sensor FOV, but is still not as high as for the KHQS catalog

Table 1 gives three measures of spatial distribution for each of the onboard catalogs. For all of the measures, a smaller value represents a more uniform distribution. The R-measure (Ref. 2) is the variance of the number of stars in the sensor FOV divided by the average number of stars in the FOV. The D-measure (Ref. 1) is the variance of the *Voronoi density* times a normalizing scale factor, where the Voronoi density is the inverse of the Voronoi cell area. The kernel of the G-measure (Ref. 1) is the average of the Delaunay cap areas. A smaller G-measure represents a distribution whose Delaunay triangles are more nearly equilateral. The D and G-measures are defined in Ref. 1, and are described briefly in Section 3.1 of the Vorosel Tool User's Manual (see the appendix). The measures of spatial distribution indicate that all three of the onboard catalogs selected using the Vorosel Tool are more uniform than the KHQS catalog.

Table 1 Measures of Spatial Distribution

Measure	Catalog				
	Candidates	KHQS	WVDR	SVDR	DCA
R	2.402	0.520	0.523	0.352	0.375
D	1.510	0.511	0.361	0.158	0.182
G	1.001	0.954	0.754	0.639	0.677

Figure 2 shows the frequency of at least k stars in the sensor FOV. The KHQS catalog has the highest frequency of three or more stars, as should be expected since the KHQS catalog contains the three brightest stars for any position of the sensor FOV. However, all three of the catalogs selected using the Vorosel Tool have a higher percentage of the sky with four or more stars. The SVDR and DCA catalogs have the highest frequency of four or more stars, but as seen in Figure 3, these catalogs also have lower frequencies of brighter stars. According to Figures 2 and 3 the WVDR catalog is very similar to the KHQS catalog: they both have nearly the same frequency of at least k stars in the sensor FOV, and the same distribution of star magnitudes. However, according to Figure 4 the WVDR catalog has fewer holes (Delaunay caps) of larger radii than the KHQS catalog. The SVDR catalog has the lowest frequency of holes with larger radii.

CONCLUSION

The observations above lead one to the following conclusions. If it is important to favor brighter stars, the weighted Voronoi density reduction is likely to give the best results. It can yield a catalog with nearly the same distribution of star magnitudes as the KHQS catalog, and nearly the same frequency of three or more stars in the sensor FOV, but will have a lower frequency of larger holes. If the star sensor accuracy does not degrade for dimmer stars, the standard Voronoi density reduction method gives the best onboard catalog. It will favor brighter stars while avoiding larger holes. Delaunay cap anchoring offers no significant advantage over the other methods.

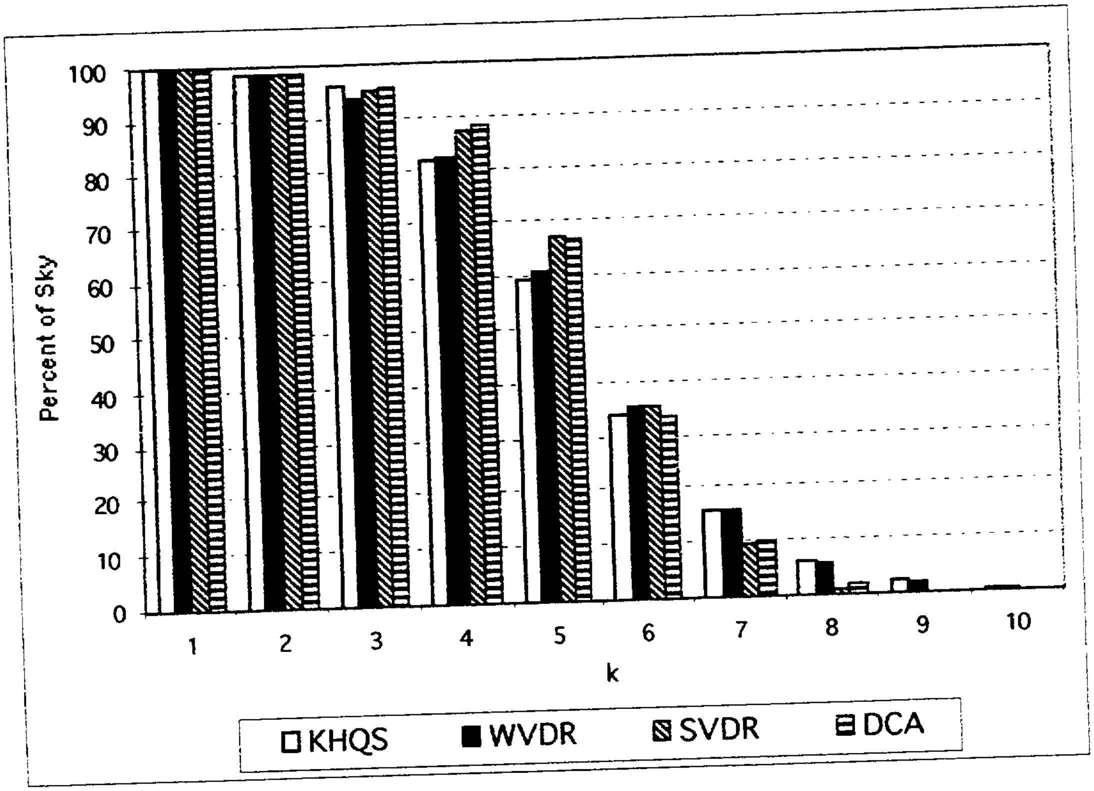


Figure 2 Frequency of at Least k Stars in the Sensor Field of View.

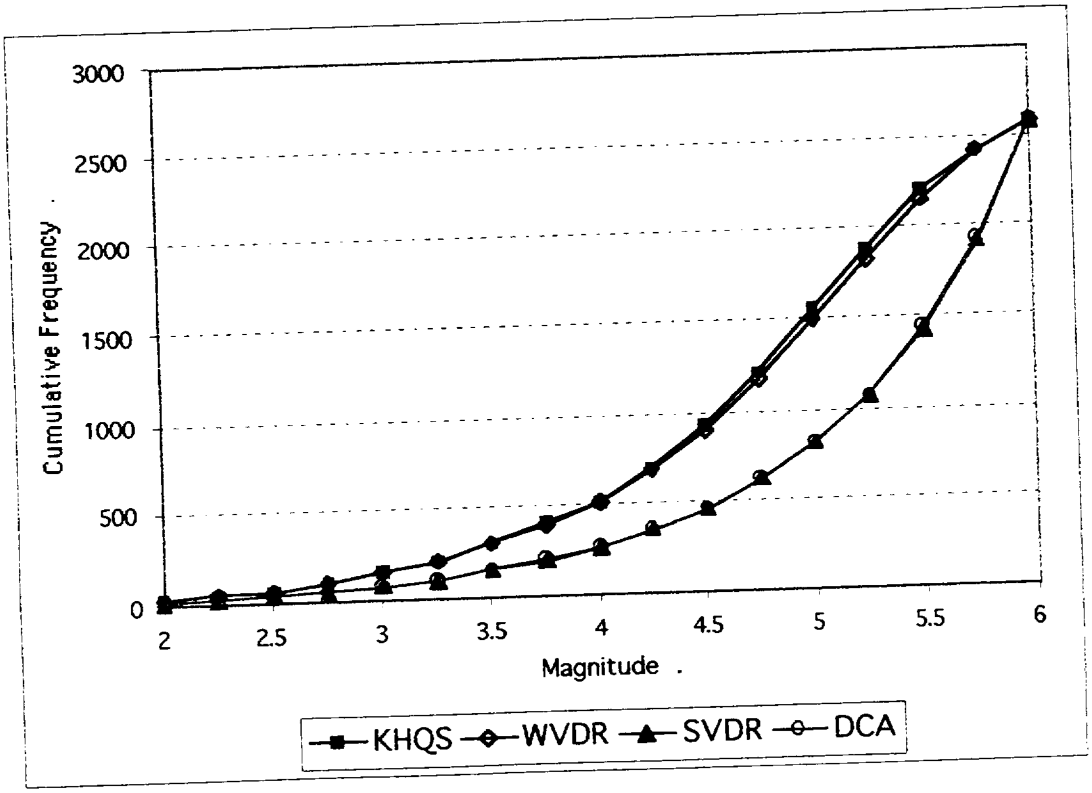


Figure 3 Cumulative Frequency of Star Magnitude.

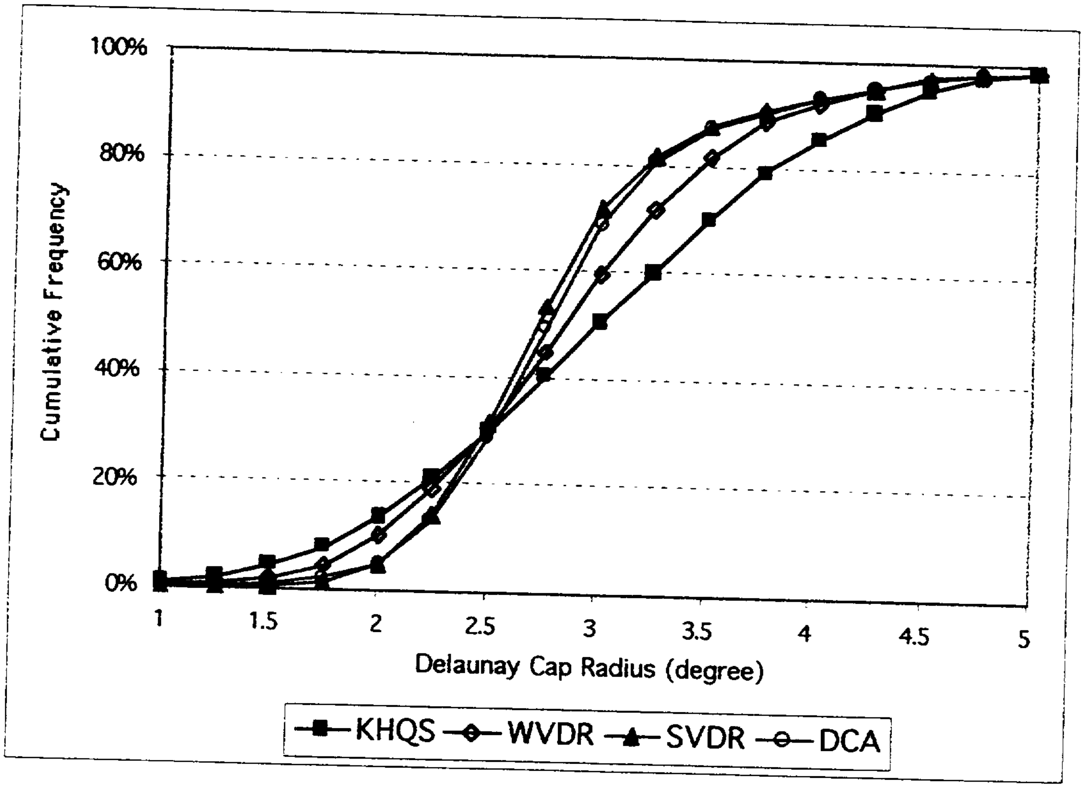


Figure 4 Cumulative Frequency of Delaunay Cap Radius.

References

1. Bauer, Robert, "Distribution of Points on a Sphere with Application to Star Catalogs", *Journal of Guidance, Control, and Dynamics*, to be published. Also in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Boston, August 1998, paper No. AIAA-98-4330.

2. Vedder, J.D., "Star Trackers, Star Catalogs, and Attitude Determination: Probabilistic Aspects of System Design," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 3, May-June 1993, pp 498-504.

3. Barry, K., Hindman, M., and Yates, R., "Application of Flight Data to Space Shuttle CCD Star Tracker Catalog Design," 16th Annual AAS Guidance and Control Conference, February 6-10, 1993, Keystone, Co.

4. Kudva, P., "Flight Star Catalog Development for EOS-AM1," NASA GSFC Contract #NAS5-32590, task 710-03, TM-421-97-008, dated 03.24.97.

Appendix:

User's Manual for the Vorosel^{TM} Tool

Version 1.0, March 1999

Written by Robert Bauer of *Bauer*Engineering Enterprises, Langhorne, Pennsylvania.

The Vorosel tool uses the Voronoi density reduction method to select stars for an onboard catalog from a larger candidate set. This manual gives an overview of the method and explains how to use the tool.

Table of Contents

List of Tables

[TM] Vorosel is a trademark of *Bauer* Engineering Enterprises, Langhorne, Pennsylvania.

1. Overview

The Vorosel tool uses the Voronoi density reduction method (Ref. 1) to select stars for an onboard catalog from a larger candidate set. The user builds a candidate set by selecting stars from a reference star catalog (such as SKYMAP) which meet certain mission and star sensor criteria, such as star magnitude, separation, proper motion, etc. The candidate stars are used as input to the Vorosel tool along with the number of stars desired for the onboard catalog. Other inputs are used to refine the star selection process as described below. The onboard catalog selected by the Vorosel tool is a very uniformly distributed sub-set of the candidates.

In this overview, the star selection process is explained, and the overall program flow of the Vorosel tool is given. Subsequent sections give more detailed explanation of the inputs and outputs, how to tailor the source code if so desired, and the meaning of error messages.

1.1 Voronoi Density Reduction

The Voronoi density reduction method to select stars for an onboard catalog is conceptually simply, yet very effective.

1) Identify a star in the densest region of the candidate set.

2) Remove the star to yield a smaller, more uniformly distributed candidate set.

3) Repeat the removal process until the desired number of stars remains.

To identify a star in the densest region of the candidate set, first the *Voronoi diagram* (Ref. 2) of the candidates is constructed. The Voronoi diagram is a subdivision of the celestial sphere into spherical polygonal cells, one for each star, so that the cell for star *P* consists of the region on the sphere closer to *P* than to any other star. The Voronoi diagram provides a natural way to identify stars in the densest region of the candidate set: the star with the smallest *Voronoi cell* area is interpreted as a star in the densest region. The Voronoi density reduction removes the star with the smallest Voronoi cell area during each iteration of the process outline above.

Several refinements of the basic Voronoi density reduction just described are possible. The simplest of these refinements is to remove the star with the smallest *weighted* Voronoi cell area, which is simply the Voronoi cell area times a weight supplied by the user. The weights can be used, for example, to favor brighter stars for the onboard catalog.

The second refinement involves the use of *anchors* supplied by the user for each star. An anchor is a multi-purpose flag: it specifies whether the respective star is in the *region of interest* for the onboard star catalog (anchor = 0 or 1), and it specifies whether the star *must be included* in the onboard star catalog (anchor = 1). If the star is not in the region of interest (anchor = −1) it is still subject to the Voronoi density reduction, but is not included in the final onboard catalog. During each iteration of the Voronoi density reduction, the star with the smallest weighted Voronoi cell area whose anchor is not 1, is removed from the candidate set. The iterations terminate when the number of candidate stars in the region of interest (anchor = 0 or 1) equals the desired number of stars for the onboard star catalog.

The use of the anchors is illustrated by the following example. Suppose the user wants to select stars for an onboard star catalog restricted to declinations above +22 degrees. Furthermore suppose that there are a handful of special guide stars which must be included in the onboard star catalog. The anchors are set as follows. The anchor for any candidate star whose declination is below +22 degrees is set to −1 signifying that it is not in the region of interest. The anchor for each of the special guide stars is set to +1 signifying that it must be included in the onboard catalog. The anchors for all other candidates, i.e. those that are in the region of interest but are not special guide stars, are set to 0 signifying that they may or may not be included in the onboard catalog depending upon the results of the Voronoi density reduction.

It may occur to the reader that if the user wants only stars in a certain region of interest, then why not simply exclude stars which are outside of this region from the candidate set. While this approach appears to be reasonable, the user should be aware of the follow nuances of the Vorosel tool. First and most serious, if the spherical cap circumscribing the candidate stars is smaller than a hemisphere, the algorithm to construct the Voronoi diagram may fail. Second, the Voronoi density reduction will tend to keep all stars along the border of the region of interest if there are no candidate stars outside the region. Thus, the star density in the onboard catalog would tend to be highest where stars are likely not to be needed (assuming the user has chosen a region of interest which is slightly larger than the expected range for star sensing). Therefore it is strongly recommended that stars outside the region of interest be included in the candidate set to act as "buffers" insuring proper construction of the Voronoi diagram, and keeping the star density uniform throughout the onboard catalog.

The final refinement of the Voronoi density reduction is call *Delaunay cap anchoring*. It is designed to limit the size of "holes" in the onboard star catalog to a specified value to the extent possible. This refinement is based on the *Delaunay triangulation* (Ref. 2) of the candidate stars. A triangulation subdivides the sky into contiguous spherical triangles whose vertices are the candidate stars. A Delaunay triangulation has the unique property that the spherical caps circumscribing these triangles are devoid of candidate stars. These " Delaunay caps" may be interpreted as holes in the candidate set. With Delaunay cap anchoring, the stars defining Delaunay caps whose radii are larger than a value supplied by the user are anchored before the Voronoi density reduction is performed. Then, during each iteration of the Voronoi density reduction, if it is found that the removal of a star from the candidate set creates a Delaunay cap larger than the limit, the star is returned to the candidate set and anchored. The user may "turn-off" Delaunay cap anchoring by supplying a limit on the cap radius of 180 degrees or greater.

There are several *run parameters* which the user must specify to control the star selection process. For example, the user must specify the number of stars to keep in the on-board catalog and whether to use weights or anchors during the Voronoi density reduction. These parameters can be supplied from the user's keyboard, or from a file. Section 2 defines the parameters and gives guidelines on how to use them to tailor the Voronoi density reduction to the user's needs.

Section 3 describes the output of the Vorosel tool. The on-board star catalog is written to a file in "SKYMAP run catalog" format (Ref. 3). A summary of results and any error messages are written to the user's monitor and to a file. Information on the Voronoi cells and Delaunay triangles for the on-board catalog are written to files for further analysis if the user so desires.

On rare occasions the user may need to modify the Vorosel tool source code to his/her specific needs. Section 4 explains how to tailor the source code for changes the user is most likely to need, including how to tailor the functions to read in the candidate stars and write out the on-board catalog.

1.2 Program Flow

The program flow for the Vorosel tool is outlined below.

1) Read in the run parameters. The run parameters are read from the file `RunParameters.txt` if it exists. If not, the user will be prompted to enter the run parameters from the keyboard.

2) Read in the candidate stars from the file `CandidateStars.txt`, and the weights and anchors from the file `StarWeights.txt`. Note that `StarWeights.txt` is needed only if the user has specified in the run parameters that weights and anchors are to be used.

3) Construct the Voronoi diagram/Delaunay triangulation (VD/DT) of the candidate stars. The VD/DT is represented by a common *quad-edge* data structure. The algorithm to construct the VD/DT is incremental and starts by building a scaffold from vertices of a regular spherical tetrahedron. The scaffold forms an initial VD/DT to start the algorithm and is removed at the end. Star in the candidate set are inserted one by one, adding and deleting edges as necessary. When all candidate stars have been inserted, the points of the scaffold are removed. See Ref. 1, 4, and 5 for details of the algorithm.

4) Measure the uniformity of the candidate star distribution as well as other statistics.

5) Perform the Voronoi density reduction to select stars for the onboard star catalog.

6) Measure the uniformity of the onboard catalog distribution as well as other statistics.

7) Write the onboard catalog to the file `OnboardStars.txt`.

8) Write the Voronoi cell area and other cell features for each star in the onboard star catalog to the file `VDGeoF.txt`. Write the Delaunay triangle area and other features for each Delaunay triangle in the onboard star catalog to the file `DTGeoF.txt`.

9) Write the edges of the Voronoi diagram/Delaunay triangulation to the file `VDDTGraph.txt`. Data in this file may be read and plotted by the Matlab script file `plotVDDT.m` supplied with the Vorosel tool.

10) Delete the data structures created and terminate the program.

Some of the actions given above may be skipped depending on the run parameters specified by the user. The formats of the inputs and outputs are given in Sections 2 and 3. Tailoring of the source code, which the user may wish to do, is explained in Section 4. Finally, error and warning messages are explained in Section 5.

2. INPUTS

The Vorosel tool inputs include the candidate stars from which the onboard catalog is selected, weights and anchors for each star should the user chose to use them, and parameters to control the execution of the tool. The following sub-sections explain each of these inputs in detail.

2.1 Candidate Stars

The candidate stars are read from the file `CandidateStars.txt`. The data for the stars is given in SKYMAP run catalog format: each line in `CandidateStars.txt` consist of 107 ASCII bytes formatted as defined in Table 1.

Table 1 SKYMAP Run Catalog Format

Bytes	Format	Description
1-9	Integer	SKYMAP Number
10-24	Floating Point	G.C.I. Unit Vector X coordinate
25-39	Floating Point	G.C.I. Unit Vector Y coordinate
40-54	Floating Point	G.C.I. Unit Vector Z coordinate
55-69	Floating Point	Instrument Magnitude
70-84	Floating Point	Proper Motion
85-92	Hexadecimal	Quality Flag
93-107	Floating Point	B-V Color Index

2.2 Run Parameters

The user may specify run parameters in the file `RunParameters.txt`. If this file does not exist the user is prompted to enter the parameters from the keyboard. An example of the contents of `RunParameters.txt` is given below. Text after a pound sign (#) to the end of the line is interpreted as a comment. The parameters are described in Table 2 and must be supplied in the order given in the table.

Example of `RunParameters.txt`

```
# Text after a pound sign (#) to the end of the line is a comment.
# Parameters must be listed in the order given below:
#
# Beginning of parameter list
2500        #    Number of stars desired for onboard star catalog
0           #    Do not use weights and anchors
180.0       #    Do not impose a Delaunay cap size limit
10.0        #    Over-sample the sky by a factor of 10
4.0         #    Sensor FOV radius in degrees
1           #    Write onboard star catalog to file
1           #    Write onboard star catalog geometric features to file
0           #    Do not write onboard star catalog VD/DT to file
# End of parameter list
```

Table 2 Run Parameters

Parameter	Units	Description
Desired size of the onboard star catalog	Unitless or Stars/Deg^2	If a number greater than 1 is given, it is interpreted as the desired number of stars for the onboard star catalog. If a number less than 1 is given, it is interpreted as the desired average star density for onboard star catalog in stars/degree^2. Note that the solid angle of a sphere is 4π steradians or $4\pi\,(180/\pi)^2$ degree^2 = 10,313.24…degree^2.
Flag to use weights and anchors	0/1	If this flag is set to 1, weights and anchors read from the file `StarWeights.txt` are used in the Voronoi density reduction method. During each iteration of the method the star with the smallest weighted Voronoi cell area is removed from the candidate list if it is not anchored. For a detailed description of how the weights and anchors are used and the `StarWeights.txt` file format see Section 2.3.
Delaunay cap radius limit	Degrees	Before the Voronoi density reduction method is applied, all candidate stars defining Delaunay caps bigger that this limit are anchored (see Section 2.3 for how the anchors are used). During the Voronoi density reduction, if the removal of a candidate star causes the creation of a Delaunay cap bigger than the limit, the star is reinserted into the candidate set and anchored. Setting the limit to 180.0 degrees (or greater) results in no stars being anchored based on Delaunay cap size.
Demographics resolution	Unitless	If the demographics resolution is greater than zero the Vorosel tool will find the frequency of k stars in the sensor field of view as the sensor bore sight is swept uniformly over the sphere. The distribution of star magnitudes will also be found. To find the frequency of having k stars in the sensor field of view, a large number of uniformly distributed sensor bore sights are generated. (The "spiral" distribution described in Ref. 1 is used.) A resolution of 10, the recommended value, means that on average 10 of the sample sensor fields of view will overlap any given point on the sphere.
Sensor field of view radius	Degrees	The sensor field of view radius is used to determine the onboard catalog demographics as described above.
Flag to write on-board catalog to file	0/1	If this flag is set to 1, the onboard star catalog will be written to the file `OnboardStars.txt`. See Section 3.2 for a description of the file format.
Flag to write VD/DT geometric features to file	0/1	If this flag is set to 1, the Voronoi Diagram and Delaunay triangulation geometrical features for the onboard star catalog will be written to the files `VDGeoF.txt` and `DTGeoF.txt` respectively. See Section 3.3 for a description of the features and file formats.
Flag to write VD/DT edges to file	0/1	If this flag is set to 1, the endpoints of the Voronoi diagram/Delaunay triangulation edges in the region of interest will be written to the file `VDDTGraph.txt`. See Section 3.4 for a description of the file format.

2.3 Star Weights and Anchors

If the flag to use weights and anchors is set (see Section 2.2) the weights and anchors are read from the file StarWeights.txt. Each line of this file contains the weight and anchor for one star. For each star read from CandidateStars.txt one line is read from StarWeights.txt. If the SKYMAP number in StarWeights.txt does not agree with the one in CandidateStars.txt an error message is issued and the Vorosel tool aborts. Table 3 defines the StarWeights.txt contents. The use of the weights and anchors by the Voronoi density reduction is explained in detail in Section 1.1.

Table 3 StarWeights.txt File Format

Parameter	Format	Description
SKYMAP Number	Integer	The SKYMAP number on the line n of StarWeights.txt must agree with the one on the line n of CandidateStars.txt. If not, an error message is issued and the Vorosel tool aborts.
Weight	Floating Point	The weight is used during the Voronoi density reduction. The star with the smallest weighted Voronoi cell area (the weight times the cell area) which is not anchored is removed from the candidate set during each iteration of the Voronoi density reduction. The weight must lie between LIGHT_WGHT_LIM and HEAVY_WGHT_LIM defined in the header file userif.h (see Section 4.1). If not, a warning message is issued and the following action is taken: If weight ≤ 0.0, the star is excluded from processing as if it were not given in the candidate star file; else if weight < LIGHT_WGHT_LIM it is set to LIGHT_WGHT_LIM; else if weight > HEAVY_WGHT_LIM the star is anchored, i.e. the anchor is set to 1 (see below).
Anchor	Integer	The anchor flag can take on four values: −2, −1, 0, and 1. The meaning of the four values is summarized below: If anchor ≤ −2 then the star is excluded from processing as if the star were not given in the candidate star file. If anchor = −1 then the star is handled by the Voronoi density reduction in the usual way, but it is not counted as a candidate for the onboard star catalog. This value is used to indicate that the star is outside the region of interest for the onboard catalog. If anchor = 0 then the star is handled by the Voronoi density reduction in the usual way and is considered a candidate for the onboard catalog. The star is written to the onboard catalog file if not removed by the Voronoi density reduction. If anchor = 1 then the star is forced to be included in the onboard catalog.

The simple bsw (build star weights) tool is supplied with the Vorosel tool to build a StarWeights.txt file. The bsw tool reads stars from CandidateStars.txt, applies formulae to find the weights and anchors based on the star features (position, magnitude, etc.), and writes the weights and anchors out to StarWeights.txt. The user must modify the source code for this tool (mainB.cc) to tailor the formulae for the weights and anchors as needed. The bsw tool is then recompiled using the makefile supplied with the Vorosel tool.

3. OUTPUTS

There are several outputs from the Vorosel tool. The most of important, of course, is the onboard catalog which is written to the file `OnboardStars.txt`. Status and statistical outputs are written to the user's monitor and to the file `RunSummary.txt`. There are several other outputs which can be used to validate the onboard catalog. These outputs include the onboard star catalog demographics, geometric features of the Voronoi diagram/Delaunay triangulation, and the edges of the Voronoi diagram/Delaunay triangulation. All of these outputs are explained in detail in the follow subsections.

3.1 Run Summary

Run summary results are written to the user's monitor and to the file `RunSummary.txt`. Included among other results are the following:

- Number of stars in the full candidate set and the number of candidate stars in the region of interest;
- Area of the region of interest;
- Uniformity measures of candidate stars in the region of interest;
- Number of stars anchored directly by the user and by adherence to the Delaunay cap size limit;
- Number of stars desired and actually selected for the onboard star catalog;
- Uniformity measures of the onboard star catalog;
- Onboard star catalog demographics;
- Error and warning messages.

The area of the region of interest is the average of the sum of Voronoi cell areas for stars in the region of interest and the sum of Delaunay triangle areas for the Delaunay triangles in the region of interest. A Delaunay triangle is considered to be in the region of interest if two of its three vertices are in the region of interest.

Two uniformity measures, the D-measure and the G-measure (Ref. 1), are calculated for the candidate stars in the region of interest and for the onboard star catalog. The D-measure equals the variance of the Voronoi density (inverse of the Voronoi cell area) over the region of interest times a normalizing scale factor. For a uniformly random distribution of stars the D-measure is statistically equal to 1.0. For a distribution wherein all Voronoi cells are of the same area, the D-measure is zero.

The kernel of the G-measure is the average Delaunay cap area divided by the ideal minimum average Delaunay cap area for Delaunay triangles in the region of interest. (A Delaunay triangle is considered to be in the region of interest if at least two of its vertices are in the region of interest.) The ideal minimum Delaunay cap area is the area of the cap circumscribing an equilateral triangle whose area equals the average Delaunay triangle area in the region of interest. A normalizing function is applied to the kernel to arrive at the G-measure. For a uniformly random distribution of stars the G-measure is statistically equal to 1.0. For a distribution wherein all Delaunay triangles are equilateral (an idealized condition), the G-measure is zero.

3.2 Onboard Star Catalog

The onboard star catalog is written to the file `OnboardStars.txt` in SKYMAP run catalog format (see Table 1). Stars appear in `OnboardStars.txt` in the same order as in `CandidateStars.txt`.

3.3 Voronoi Diagram/Delaunay Triangulation Geometric Features

The geometric features of the Voronoi diagram are written to the file `VDGeoF.txt` if the flag to write the features is set (see Section 2.2). These features include: the Voronoi cell area; the Voronoi cell perimeter length; and the Voronoi cell aspect number, which represents how close the Voronoi cell is to being a regular hexagon. The aspect number A_{Vk} is given by the following formula:

$$A_{Vk} = ln\left(\lambda_k / \gamma_k\right)$$

where

$\lambda_k =$ Length of the k^{th} Voronoi cell perimeter

$\gamma_k =$ Perimeter length of a regular spherical hexagon whose area is the same as the k^{th} Voronoi cell

Each line of VDGeoF.txt contains the features for one Voronoi cell. The file format is given in Table 4.

Table 4 VDGeoF.txt File Format

Bytes	Format	Description
1-9	Integer	SKYMAP Number of the star in the Voronoi cell
10-24	Floating Point	Voronoi cell area in degree^2
25-39	Floating Point	Voronoi cell perimeter length in degree
40-54	Floating Point	Voronoi cell aspect number

The geometric features of the Delaunay triangulation are written to the file DTGeoF.txt if the flag to write the features is set (see Section 2.2). These features include: the Delaunay triangle area; the Delaunay cap radius; and the Delaunay triangle aspect number, which represents how close the Delaunay triangle is to being equilateral. The aspect number A_{Dk} is given by the following formula:

$$A_{Dk} = ln(\alpha_k / \beta_k)$$

where

α_k = area of the k^{th} Delaunay cap

β_k = area of a cap circumscribing an equilateral spherical triangle whose area is the same as the k^{th} Delaunay triangle

Each line of DTGeoF.txt contains the features for one Delaunay triangle. The file format is given in Table 5.

Table 5 DTGeoF.txt File Format

Bytes	Format	Description
1-9	Integer	SKYMAP Number of the star at the first vertex of the Delaunay triangle
10-18	Integer	SKYMAP Number of the star at the second vertex of the Delaunay triangle
19-27	Integer	SKYMAP Number of the star at the third vertex of the Delaunay triangle
28-42	Floating Point	Delaunay triangle area in degree^2
43-57	Floating Point	Delaunay cap radius in degree
58-72	Floating Point	Delaunay triangle aspect number

3.4 Voronoi Diagram / Delaunay Triangulation Edges

The endpoints of the Voronoi diagram / Delaunay triangulation edges in the region of interest are written to the file VDDTGraph.txt if the flag to write the edges is set (see Section 2.2). Each line in VDDTGraph.txt gives the endpoints for one Voronoi diagram edge and the dual edge in the Delaunay triangulation. The file format is given in Table 6.

Table 6 VDDTGraph.txt File Format

Bytes	Format	Description
1-10	Floating Point	X coordinate of the first endpoint of the Voronoi Edge
11-20	Floating Point	Y coordinate of the first endpoint of the Voronoi Edge
21-30	Floating Point	Z coordinate of the first endpoint of the Voronoi Edge
31-40	Floating Point	X coordinate of the second endpoint of the Voronoi Edge
41-50	Floating Point	Y coordinate of the second endpoint of the Voronoi Edge
51-60	Floating Point	Z coordinate of the second endpoint of the Voronoi Edge
61-70	Floating Point	X coordinate of the first endpoint of the Delaunay Edge
71-80	Floating Point	Y coordinate of the first endpoint of the Delaunay Edge
81-90	Floating Point	Z coordinate of the first endpoint of the Delaunay Edge
91-100	Floating Point	X coordinate of the second endpoint of the Delaunay Edge
101-110	Floating Point	Y coordinate of the second endpoint of the Delaunay Edge
111-120	Floating Point	Z coordinate of the second endpoint of the Delaunay Edge

4. Tailoring the Source Code

This section explains those parts of the source code which the user may wish to tailor to his/her specific needs. This code in contained in two files, userif.h and userif.cc. The source code may be recompiled using the makefile included with the Vorosel tool.

4.1 Global Parameters

The header file userif.h defines eight global parameters. A description of each is given below. If the user changes the values of these parameters the Vorosel tool must be recompiled.

CATNAME = "SKYMAP" is the name of the source catalog. It is used in printing diagnostic information. The Vorosel tool assumes the candidate stars are from a SKYMAP run catalog. The character string assigned to CATNAME may be changed if the source of the candidate stars is changed. (Such a change might also require that the function to read the candidate stars be changed as described in Section 4.)

LINE_LENGTH = 107 is the length of a line in CandidateStars.txt. It is used to check the validity of star data read in from CandidateStars.txt. The Vorosel tool assumes the candidate stars are from a SKYMAP run catalog. The user may need to change the value of LINE_LENGTH if the format of CandidateStars.txt is changed. (Such a change would also require that the function to read the candidate stars be changed.)

COIN = 1.0e-6 (0.2 arcseconds) is used in the test of coincidence of star locations. If two stars in the candidate catalog file are within COIN radians of each other, only one of the two is retained in the Delaunay triangulation as decided by a call to BetterStar(). See Section 4.4 for a description of the function BetterStar(). The value assigned to COIN may be change, but it must be greater than or equal to the square root of EPS2.

LIGHT_WGHT_LIM = 1.0e-3 and HEAVY_WGHT_LIM = 1.0e+3 are used to limit the star weights. If a weight is less than 0, a warning is issued and the star is excluded from processing as if it were not given in the candidate star file. If a weight is between 0.0 and LIGHT_WGHT_LIM, a warning is issued and the weight is set to LIGHT_WGHT_LIM. If a weight is greater than HEAVY_WGHT_LIM, a warning message is issued and the star's anchor is set to +1, signifying that that is must be included in the onboard catalog. (See Sections 1.1 and 2.3.) If the user wishes to change the values of LIGHT_WGHT_LIM or HEAVY_WGHT_LIM, their quotient must be made greater than or equal to 1000 times the machine epsilon.

TALLY_LIMIT = 31 is used to limit the number of stars counted in the sensor field-of-view when finding the onboard star catalog demographics. It is also used to limit the recursion of the demographics workhorse Knock(). If TALLY_LIMIT is increased, there must be enough core memory to allow as many as TALLY_LIMIT levels of recursion by Knock().

BIN_SIZE = 0.5 is the size of the bins used to find the star magnitude histogram for the onboard catalog.

MARKPERIOD = 50 is used to issue progress marks during the Voronoi density reduction. To increase the frequency of marks, decrease MARKPERIOD.

EPS1 = 1.0e–7 is used to check the length of each candidate star position vector. If the length is different from 1.0 by more than half EPS1, a warning is issued and the position vector is normalized. There should rarely be a reason to change the value of EPS1. If it is changed it must be greater than or equal to the value of EPS2 and the machine epsilon.

EPS2 = 2.0e–14 is used for the following checks:

1) When triangulating the cavity created by a star removal during the Voronoi density reduction, if the triangulation cannot be made strictly Delaunay, encroachment as great as half EPS2 is allowed by stars into the caps defined by the triangles in the triangulation of the cavity. If it is still not possible to triangulate the cavity, an error message is issued and the Vorosel tool aborts.
2) Whether the triangulation of the stars is Delaunay. If any star encroaches into a Delaunay cap by more than 1.1 times EPS2, a warning message is issued and the Vorosel tool aborts.
3) Whether the sum of Voronoi cell areas is 4π. If the sum divided by 4π is different from 1.0 by more than EPS2, a warning message is issued and the Vorosel tool aborts.
4) Whether the sum of Delaunay triangle areas is 4π. If the sum divided by 4π is different from 1.0 by more than EPS2, a warning message is issued.
5) Whether two vectors are equal. Two vectors are considered equal if the magnitude squared of their difference is less than EPS2.

There should rarely be a reason to change the value of EPS2. If it is changed it must be greater than or equal to 50 times the machine epsilon.

4.2 Read Stars Function

The function ReadStars() in the source file userif.cc reads the candidate stars from the file CandidateStars.txt. It is designed to read star features in SKYMAP run catalog format (see Table 1). The user may wish to modify this code to read other formats. Changes to ReadStars() should be restricted to that part of the code delineated by // USER SPECIAL //.

4.3 Write Stars Function

The function WriteStars() in the source file userif.cc writes the onboard star catalog to the file OnboardStars.txt. It is designed to write star features in the same format as found in CandidateStars.txt. The user may wish to modify this code to write other formats. Changes to WriteStars() should be restricted to that part of the code delineated by // USER SPECIAL //.

4.4 Better Star Function

The function BetterStar() in the source file userif.cc is called when two candidate stars are found to occupied the same site on the sphere. BetterStar() determines which of the two stars is better. The better star is kept as a candidate and the other star is discarded. As delivered, BetterStar() simply chooses the star with the smaller star number as the better star. The user may with to change this functions. If so, changes should be restricted to that part of the code delineated by // USER SPECIAL //.

5. Error and Warning Messages

The Vorosel tool performs internal diagnostics for proper program execution. If a problem is detected and corrective action is possible, it issues a WARNING message explaining the nature of the problem and what action is being taken. It then proceeds with normal execution. The user should scrutinize the results carefully to determine adequacy for the intended purpose. If a problem is detected and corrective action is not possible, an ERROR message is issued and the Vorosel tool attempts to gracefully abort. Any results should be held suspect. The following problems will cause a WARNING or ERROR message.

1) The parameters COIN, EPS1, EPS2, LIGHT_WGHT_LIM, and HEAVY_WGHT_LIM are not consistent with each other or the machine epsilon per Section 4.1. The tool issues an ERROR message and aborts.

2) CandidateStars.txt does not exist. The tool issues an ERROR message and aborts.

3) A line in CandidateStars.txt is shorter than LINE_LENGTH characters. The tool issues an ERROR message and aborts.

4) StarWeights.txt does not exist and the user has specified weights and anchors are to be used. The tool issues a WARNING message and proceeds without using weights and anchors.

5) Star numbers in CandidateStars.txt do not agree with those in StarWeights.txt. The tool issues an ERROR message and aborts.

6) Star position vector is not a unit vector. The tool issues a WARNING message, normalizes the vector, and proceeds with execution.

7) Star weight is not between LIGHT_WGHT_LIM and HEAVY_WGHT_LIM. The tool issues a WARNING message, takes corrective action as explained in Table 2, and proceeds with execution.

8) There are less than four valid stars in CandidateStars.txt. The tool issues an ERROR message and aborts.

9) Two or more candidate stars are co-located, i.e., their unit vectors are within COIN radians of each other. The tool issues a WARNING message and calls BetterStar() to decide which star to keep. For further information regarding the parameter COIN and the function BetterStar(), see Sections 4.1 and 4.4 respectively.

10) The area of the largest Delaunay cap among the candidate stars is greater than 2π, a condition the algorithm used by the Vorosel tool may not be able to accommodate. If such is the case, the tool will fail to removed the scaffold used to construct the Voronoi diagram/Delaunay triangulation of the candidate stars. In response to this failure, the tool issues an ERROR message and aborts.

Other diagnostics are performed on the integrity of the internal data structures (the linked list of stars, the linked list of Voronoi vertices, and the quad-edge data structure representing the Voronoi diagram/Delaunay triangulation). None of these diagnostics should produce an ERROR message unless unsound modifications have been made to the source code.

References

1. Bauer, Robert, "Distribution of Points on a Sphere with Application to Star Catalogs", *Journal of Guidance, Control, and Dynamics*, to be published.

2. Preparata, F.P., and Shamos, M.I., *Computational Geometry*, ©1985 Springer-Verlag New York Inc., ISBN 0-387-96131-3, pp. 19, 56, 204-223.

3. Tracewell, David, "Star Catalogs", an internet world wide web page, Goddard Space Flight Center, Flight Dynamics Branch.

4. Guibas, L. and Stolfi, J., "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams," ACM Transactions on Graphics, Vol. 4, No. 2, April 1985, pp. 74-123.

5. Lischinski, D., "Incremental Delaunay Triangulation," in *Graphics Gems IV*, Paul S. Heckbert ed., ©1994 Academic Press Inc., ISBN 0-12-336155-9, pp. 47-59.

Improved Earth Sensor Performance Using a Sequentially Correlated Noise Model[*]

J. Sedlak
Computer Sciences Corporation
Lanham-Seabrook, MD, USA 20706

Abstract

Spacecraft attitude estimation by means of an extended Kalman filter requires a reasonably true model of the inherent noise of each sensor. For some sensors, the largest uncorrected noise comes not from the sensor itself, but from errors in the model for the predicted observations. This is certainly the case for Earth horizon sensors. The Earth horizon as seen from low Earth orbit is nearly a circle whose radius depends primarily on altitude. A straightforward latitude-dependent correction is added to this to account for the oblateness of the Earth. There also are both seasonal and stochastic variations in the horizon height. The seasonal variations can be predicted to some limited degree based on models derived from historical data. The stochastic component characteristically shows variations that are correlated both in time and space but which are unpredictable over long time spans.

This work investigates whether Earth horizon sensor performance can be improved by solving for its systematic error as an augmentation of an attitude Kalman filter. It is found that using only Earth and Sun sensors, the augmented state is not fully observable. Even when magnetometer data is included, only the pitch axis component of the error can be improved; the roll component is unobservable.

1. Introduction

Many low orbit Earth science missions use Earth sensors either as the primary attitude determination sensors or for initial attitude acquisition prior to stepping up to star tracker-based control. Earth sensors provide direct information to the onboard control system about the spacecraft attitude relative to the Earth's surface. This is clearly an advantage for Earth-oriented platforms. However, Earth sensors are subject to errors that prevent them from serving as primary sensors on some missions. Earth imaging missions, for example, require both high accuracy and platform stability while multi-spectral views are being scanned. In this case, star trackers usually are used as the primary sensors in combination with gyroscopes packaged as a three-axis Inertial Reference Unit (IRU).

Earth sensors detect the Earth limb at two or more horizon points. These measurements can be expressed as an Earth-chord angular width and a rotation of the chord center with respect to the spacecraft body. Knowing the spacecraft altitude, these then yield a body-frame observation of the nadir vector. Thus the observed nadir is essentially the center of the Earth horizon as seen from orbit. This measurement (or its corresponding reference vector) must be corrected for deviations of the true Earth horizon from a nadir-centered circle. A simple, latitude-dependent function corrects for the Earth's equatorial bulge. More difficult to compensate are the actual variations in the horizon height as seen by the Earth

[*] This work was supported by the National Aeronautics and Space Administration (NASA) / Goddard Space Flight Center (GSFC), Greenbelt, MD, Contract GS-35F-4381G, Task Order no. S-24280-G.

NASA/GSFC, Guidance, Navigation and Control Center, *Flight Mechanics Symposium*, Greenbelt, MD USA, May 1999.

limb detectors. These detectors are sensitive to infrared radiation in a narrow passband. This passband is chosen to see the atmosphere at wavelengths where it has the least day/night and seasonal variability, however a certain amount of variability is still present. These horizon height variations constitute nadir observation errors of up to a few tenths of a degree for low Earth orbiting spacecraft.

While some progress has been made at using historical data for predicting and correcting for repeatable, seasonal atmospheric variations (References 1, 2, and 3), there remains an as yet unpredictable component. This may be partly periodic, but not simply seasonal, and partly stochastic, such as large-scale blooms of relatively warm air into the stratosphere which perturb the isotherms to which the detector is sensitive (very roughly 200 K at 40 km). It is this sort of error that precludes the use of Earth sensors as primary attitude sensors on platforms with moderately high precision pointing requirements.

This paper investigates whether the performance of Earth horizon sensors can be improved by estimating the horizon height variations as part of a filter for the attitude. The additional terms in an augmented state vector provide roll and pitch corrections to the reference nadir vector, known from the spacecraft ephemeris. The noise model is taken to be a 1st-order Gauss-Markov process. This relatively simple model gives the sensor corrections an autocorrelation that decays exponentially in time. The filter must also solve for the IRU biases to be useful.

Test applications of the filter show that small improvements can be obtained on one axis of the Earth sensor (this is the pitch axis about which the spacecraft rotates at one revolution per orbit). However, the filter could not reduce the error about the second axis (the roll axis). Further study showed that this system can be proven to be unobservable.

Section 2 of this paper gives a derivation of an extended Kalman filter (EKF) for estimating the attitude, the IRU biases, and the Earth sensor correction. Section 3 presents results from test cases using both simulated data and flight data from the Total Ozone Mapping Spectrometer-Earth Probe (TOMS-EP). Section 4 discusses the problems with the EKF and investigates the observability conditions. Section 5 gives discussion and conclusions.

2. Earth Sensor Error Filter

An EKF for Earth sensor systematic errors has been implemented as a modification of the Unit Vector Filter (UVF) described in Reference 4. The UVF ancestry can be traced to algorithms given in Reference 5. The approach taken here for Earth sensor error estimation is very similar to that described in References 6 and 7 for estimating systematic errors in the geomagnetic field used as the reference for magnetometer based attitude determination.

Figure 1 shows Earth sensor pitch and roll measurements from TOMS-EP spanning 8 orbital periods. The data already have been corrected for Earth oblateness. Two types of remaining errors can be seen. There are random errors that seem to be inherent to the sensor and which are adequately represented by Gaussian distributed white noise. There are also systematic errors that this filter is designed to compensate. These may occur randomly but persist over a significant fraction of an orbit, or they may be periodic with frequency components up to a few times the orbital rate. The estimates of these sensor correction terms are included with the attitude quaternion and, optionally, the IRU biases as the augmented state vector. The systematic errors presumably derive from variations in the Earth horizon radiance and are modeled as exponentially-correlated random variables. A single time constant and an

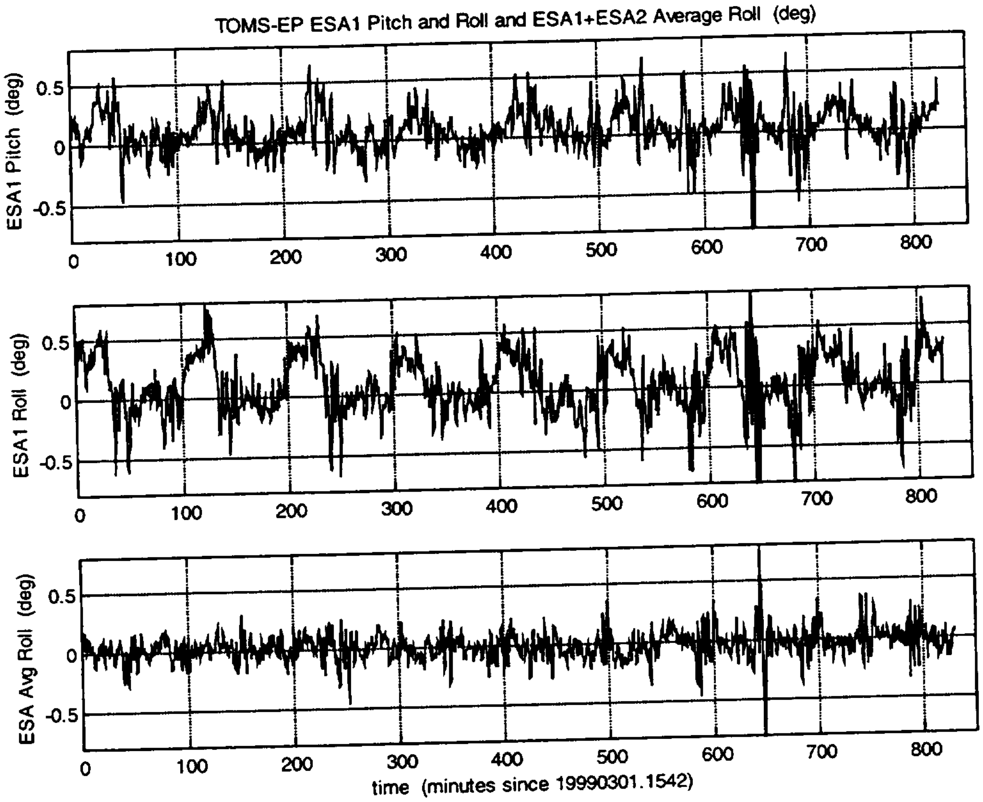


Figure 1. Earth sensor pitch and roll measurements from TOMS-EP. The top two plots show pitch and roll from sensor 1, and the bottom plot shows the averaged roll angle from sensors 1 and 2.

initial uncertainty are all that are needed for each axis. Also, since the noise sources are physically distinct for the sensor and the model errors, the white and correlated noise sequences are taken to have no cross-correlation. Given the periodic sensor errors, a noise model with a decaying *sinusoidal* autocorrelation would match the measurements better. However, earlier work on estimating errors in the geomagnetic field model has shown that this added complexity can be expected to improve the attitude determination only slightly (Reference 7).

There are two Earth sensors on TOMS-EP. These sensors detect the intersection of the horizon with scan cones centered on the positive and negative y-axis (that is, on the right and left side of the spacecraft, taking the direction of flight as forward). The bottom plot in Figure 1 shows the advantage of averaging these two Earth sensors. With this left/right geometry, the roll errors are significantly reduced. However, averaging the two sensors does not improve the pitch measurements.

The filter described below follows trends in Earth sensor residuals (that is, measured minus predicted observation vectors). The correction estimated at a given sensor update decays with time until the next update. This represents the likelihood that the residual at one moment is predictive of the next. The actual residual need not decay exponentially, but the estimate of it does. This type of filter was first derived by Bryson and Henrikson (Reference 8).

Let x_i represent the unaugmented error state vector at time t_i. The filter estimates the error state to avoid complications from the attitude quaternion normalization constraint in the full state (details are given in Reference 5). The components of x_i can be divided into the attitude error, α, and IRU bias error, Δb. The state is propagated from t_i to t_{i+1} according to

$$x_{i+1} = \Phi_i x_i + w_i \tag{1}$$

where Φ_i makes use of data from the IRU, corrected for the bias, and where w_i is a vector of zero-mean white noise sequences driving random walks in the accumulated angle and in the IRU bias. In a continuous-time model, the corresponding white noise source is taken to have spectral density

$$Q(t) = \begin{bmatrix} \sigma_{att}^2 I & 0 \\ 0 & \sigma_{bias}^2 I \end{bmatrix} \tag{2}$$

where I is the 3x3 identity matrix. See References 4 or 5 for the transition matrix Φ_i and for discussion of the IRU model and the relationship between $Q(t)$ and its discrete-time counterpart, Q_i.

The exponentially correlated error term for correcting the Earth sensor is ε_i. This can be a 2-vector representing the pitch and roll corrections. In a continuous-time model, the evolution equation for $\varepsilon(t)$ is

$$\dot{\varepsilon}(t) = -\beta\varepsilon(t) + u(t) \tag{3}$$

The correlation time is $\tau = 1/\beta$. From Eq. (3), the discrete-time propagation is

$$\varepsilon_{i+1} = \Psi\varepsilon_i + u_i \tag{4}$$

with

$$\Psi_i = e^{-\beta\Delta t_i} I \tag{5}$$

$$q_i = E\left[u_i u_i^T\right] \tag{6}$$

$$E\left[u(t_1)u^T(t_2)\right] = q(t_1)\delta(t_1 - t_2) \tag{7}$$

The spectral density matrix, $q(t)$, is assumed constant. Let p be the variance of $\varepsilon(t)$, representing a measure of the uncertainty in the systematic error in the sensor residuals. In the absence of filter updates, take p to be a constant equal to $\sigma^2_{corr} I$. In order for $dp/dt = 0$, one must have $q = 2\beta p$. Then, evaluating Eq. (6) leads to

$$q_i = \sigma_{corr}^2\left(1 - e^{-2\beta\Delta t_i}\right)I \tag{8}$$

Now, the augmented error state, x^a_i, can be written as

$$x_i^a = \begin{bmatrix} x_i \\ \varepsilon_i \end{bmatrix} \tag{9}$$

The propagation equations for the augmented state and its covariance, P^a, are

$$x_{i+1}^a = \Phi_i^a x_i^a + \begin{bmatrix} w_i \\ u_i \end{bmatrix} \tag{10}$$

$$\Phi_i^a = \begin{bmatrix} \Phi_i & 0 \\ 0 & \Psi_i \end{bmatrix} \tag{11}$$

$$P_{i+1}^a = \Phi_i^a P_i^a \Phi_i^{aT} + Q_i^a \tag{12}$$

$$Q_i^a = \begin{bmatrix} Q_i & 0 \\ 0 & q_i \end{bmatrix} \tag{13}$$

The sensor residual, y_i, for any of the attitude sensors is given as

$$y_i = H_i x_i + v_i \tag{14}$$

where the sensitivity matrix, H_i, models the linear sensor response to state changes, and where the inherent sensor noise v_i is Gaussian distributed, zero-mean, and white with covariance

$$R_{ES} = E\left[v_i v_i^T\right] = \sigma_{ES}^2 I \tag{15}$$

for the Earth sensor (ES). Also, $R_{DSS} = \sigma^2_{DSS} I$ for the digital Sun sensor (DSS), and similarly for any other attitude sensors. The observation model for H_i is constructed by expanding the residual to first-order in the error state, as follows. Let V_{ref} be the inertial frame reference nadir vector obtained from the spacecraft ephemeris. (All the following expressions carry a time subscript i that is suppressed for convenience.) The inertial frame unit vector is rotated into the exact body frame as

$$B_{ref} = e^{[\alpha\times]} A_{est} V_{ref} \tag{16}$$

where A_{est} is the a priori estimate of the attitude matrix and

$$[\alpha\times] \equiv \begin{pmatrix} 0 & -\alpha_z & \alpha_y \\ \alpha_z & 0 & -\alpha_x \\ -\alpha_y & \alpha_x & 0 \end{pmatrix} \tag{17}$$

where α is the unknown correction to the estimated attitude needed to rotate it to the true state (the same α as in the error state in Eq. (1)). The actual measured body frame vector then is taken to be

$$B_{meas} = e^{[\alpha\times]} A_{est} V_{ref} + \varepsilon + v \tag{18}$$

The ε correction is written here simply as a body frame bias. Similar results are obtained if the correction is expressed as a rotation in either the body or orbital frame. The exact form of the sensitivity and transition matrices will change, but for small angle corrections the resulting filter has the same effect. For sensors other than the Earth sensor, one drops ε from Eq. (18). The estimated body frame measurement vector is

$$B_{est} = A_{est} V_{ref} + \varepsilon_{est} \tag{19}$$

Expanding the residual, y, one finds

$$\begin{aligned} y &= B_{meas} - B_{est} \\ &\approx -\left[A_{est} V_{ref} \times\right] \alpha + \Delta\varepsilon + v \end{aligned} \tag{20}$$

where $\Delta\varepsilon = \varepsilon - \varepsilon_{est}$ is the correlated noise component of the error state. Thus, the Earth sensor sensitivity matrix for the augmented state is

$$H^{a}_{ES} = \left[\; -\left[A_{est} V_{ref} \times\right] \quad 0_{3\times3} \quad I_{3\times2} \right] \tag{21}$$

where

$$I_{3\times2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{22}$$

and the sensor noise covariance reduces to the uncorrelated part, R_{ES}, from the term v. For other attitude sensors such as the DSS or three-axis magnetometer (TAM), one has, e.g.,

$$H^{a}_{DSS} = \left[\; -\left[A_{est} V_{ref} \times\right] \quad 0_{3\times3} \quad 0_{3\times2} \right] \tag{23}$$

3. Results

The filter has been tested using both simulated data and flight data from TOMS-EP. The parameters available for tuning the filter are the correlation time, τ, the Earth sensor noise parameters, σ_{ES} and σ_{corr}, the Sun sensor error, σ_{DSS}, and the IRU noise sources in Eq. (2). A wide range of values has been used in tests; the filter is not highly sensitive to changes in the tuning. The results presented here take the correlation time to be 1500 sec (1/4-orbit) and $\sigma_{corr} = 0.1$ deg. The inherent Earth sensor noise is taken as $\sigma_{ES} = 0.15$ deg. The Sun sensor noise is $\sigma_{DSS} = 0.25$ deg. The simulated IRU noise is $\sigma^2_{att} = 10^{-12}$ rad^2/sec and $\sigma^2_{bias} = 4.6 \times 10^{-20}$ rad^2/sec^3. The IRU noise was increased to $\sigma^2_{att} = 10^{-10}$ rad^2/sec for the TOMS-EP tests because of the coarseness of the available rate data.

The attitude and orbit scenario and Sun visibility for the simulations were very similar to the actual TOMS-EP data. The spacecraft is Earth-oriented in a near-circular, Sun-Synchronous orbit with an inclination of 98.4 deg, period of 99.6 min, and an 11 a.m. local time ascending node. It has two Sun sensors, fore and aft, so Sun data are available for a brief period over both the North and South Pole. This provides yaw angle and yaw IRU bias observability.

The advantage of using simulated data is that the true attitude and sensor perturbations are known. TOMS-EP does not carry a high precision sensor such as a star tracker, so its onboard computer (OBC) attitude estimate has an uncertainty of roughly the same size as the performance goals of the correlated noise filter. More importantly, the OBC attitude is subject to the same systematic errors this filter is attempting to remove, so direct comparison with the OBC only demonstrates the consistency between the OBC attitude and the erroneous data used by the OBC for control. For this reason, attitude comparisons only from the simulated data are presented. The TOMS-EP flight data are used mainly to demonstrate that the filter is able to remove most systematic errors from the sequence of residuals even with poor IRU propagation and with sensor noise varying widely from the Poles to the Equator.

In the first test, a standard EKF was used to estimate the attitude and IRU biases using over 8 orbits of TOMS-EP Earth sensor, DSS, and IRU data. This EKF was used only for performance comparison and does not account for any noise correlations. A single effective Earth sensor is created from the average of the left and right sensors, as discussed in Section 2. The residuals obtained from this test are shown in Figure 2. Systematic oscillations in the residuals can be seen. These are driven by the systematic sensor errors seen previously in Figure 1.

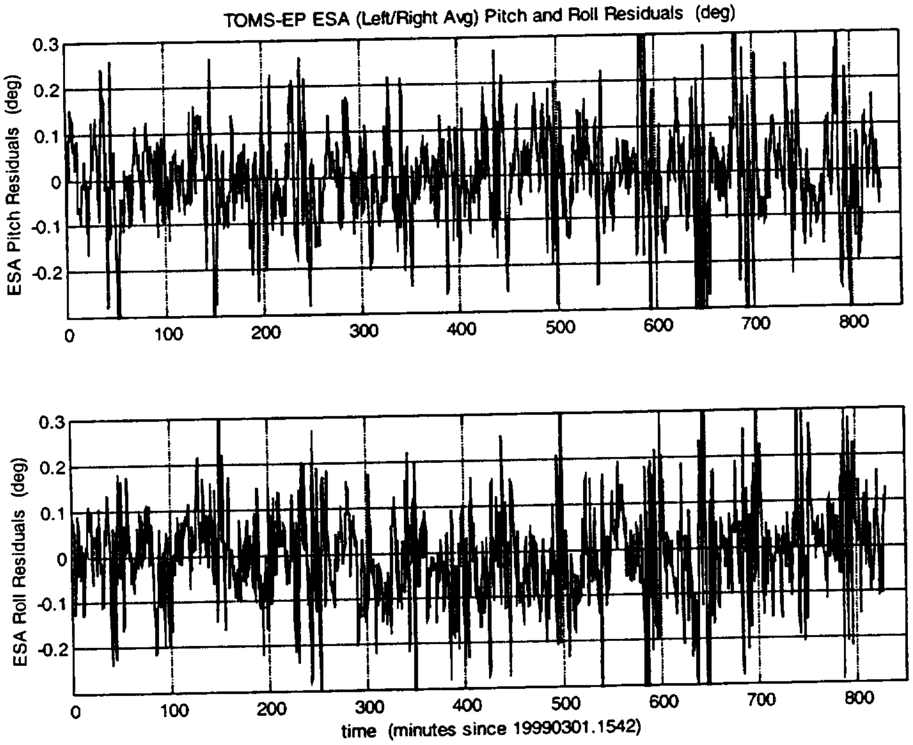


Figure 2. TOMS-EP Earth sensor pitch and roll residuals from a standard EKF. This filter does not attempt to compensate for systematic error.

The same TOMS-EP data next was used with the correlated noise filter. The residuals are given in Figure 3. This should be compared with the standard EKF residuals in Figure 2. The filter has successfully removed most apparent periodicities and systematic errors. These errors, shown in Figure 4, have been estimated and used to compensate the Earth sensor observations.

All the remaining tests described here were performed with simulated data. The noise characteristics were all chosen to approximate the TOMS-EP example. Results for the sensor residuals from the EKF and the correlated noise filter are similar to those shown in Figures 2 and 3. In fact, the residuals sequence after passing the simulated data through the correlated noise filter is nearly indistinguishable from white noise (Figure 5).

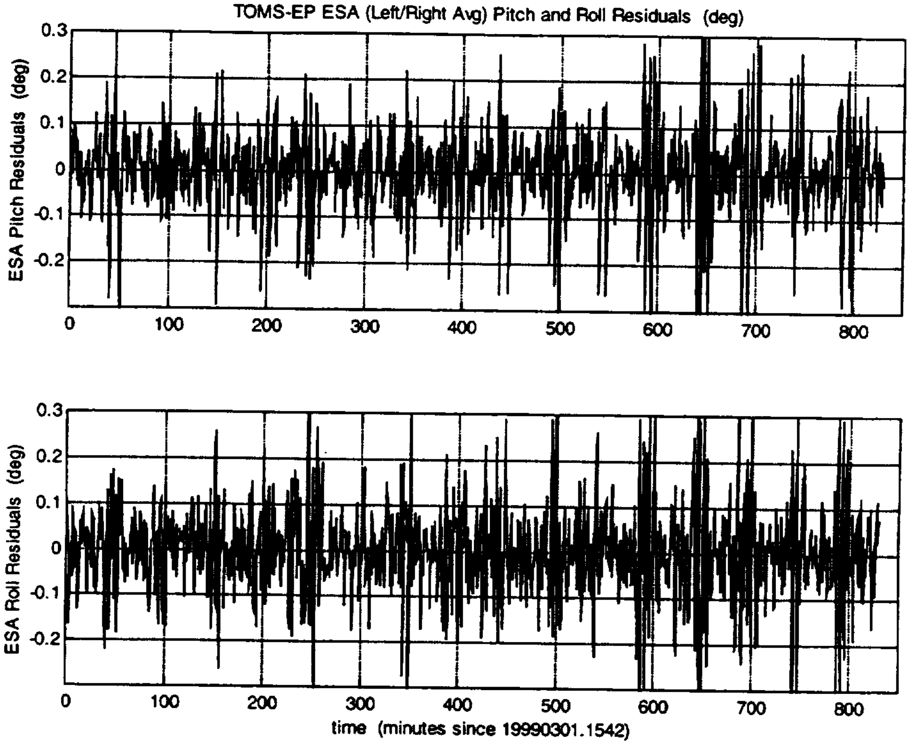


Figure 3. TOMS-EP Earth sensor pitch and roll residuals from the correlated noise filter.

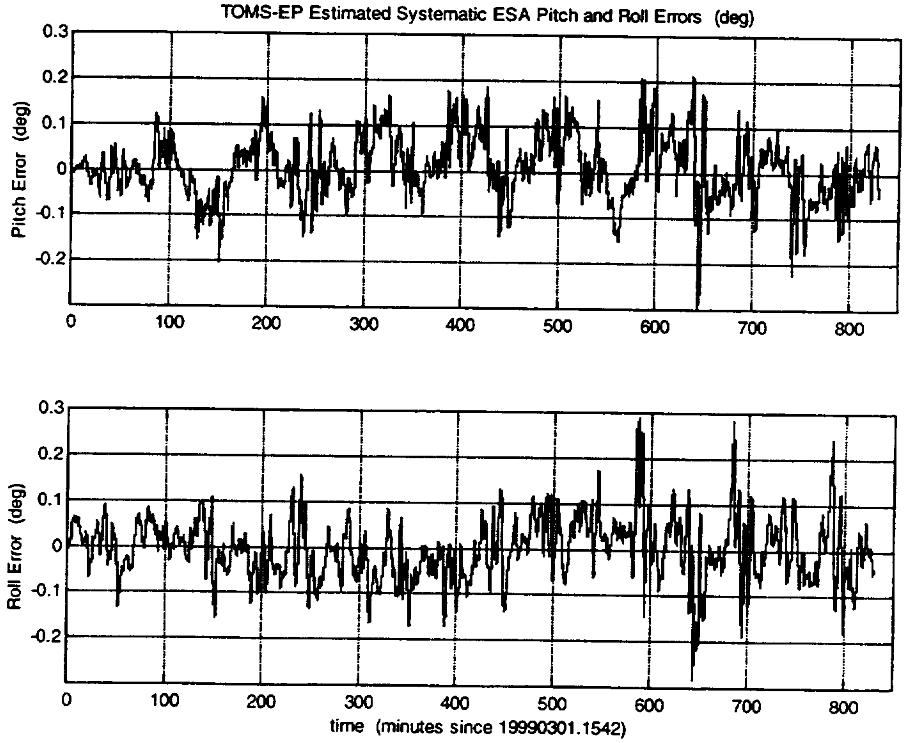


Figure 4. Estimate of the TOMS-EP Earth sensor correlated noise.

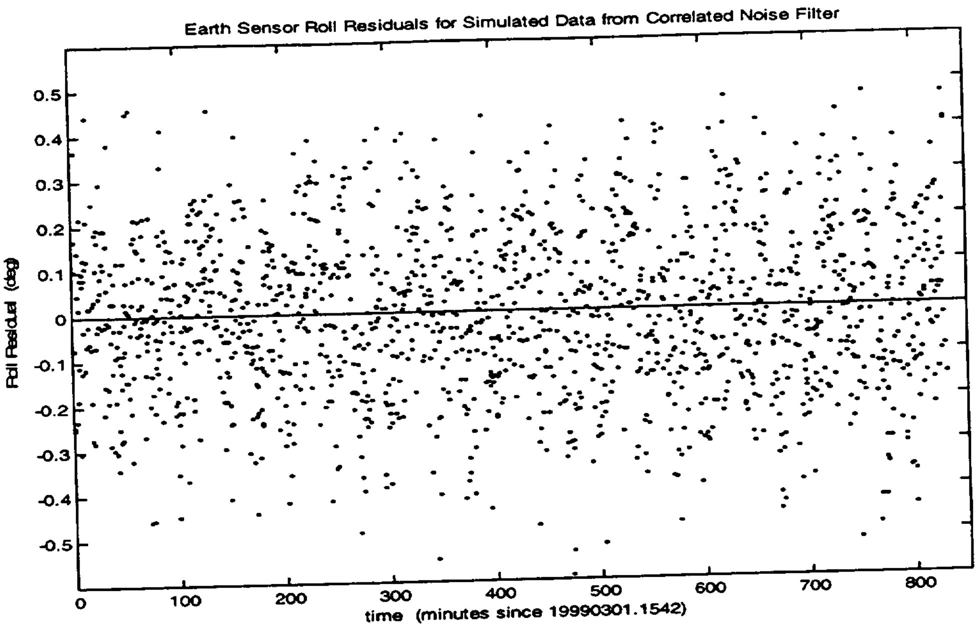


Figure 5. Earth sensor roll residuals from the correlated noise filter with simulated data.

Next, attitudes from both filters are compared with the truth model. These comparisons are given in Figure 6. The figure shows the error rotation angles from the true to the estimated attitude. These are solid lines for the correlated noise filter and dashed lines for the standard EKF. It is immediately

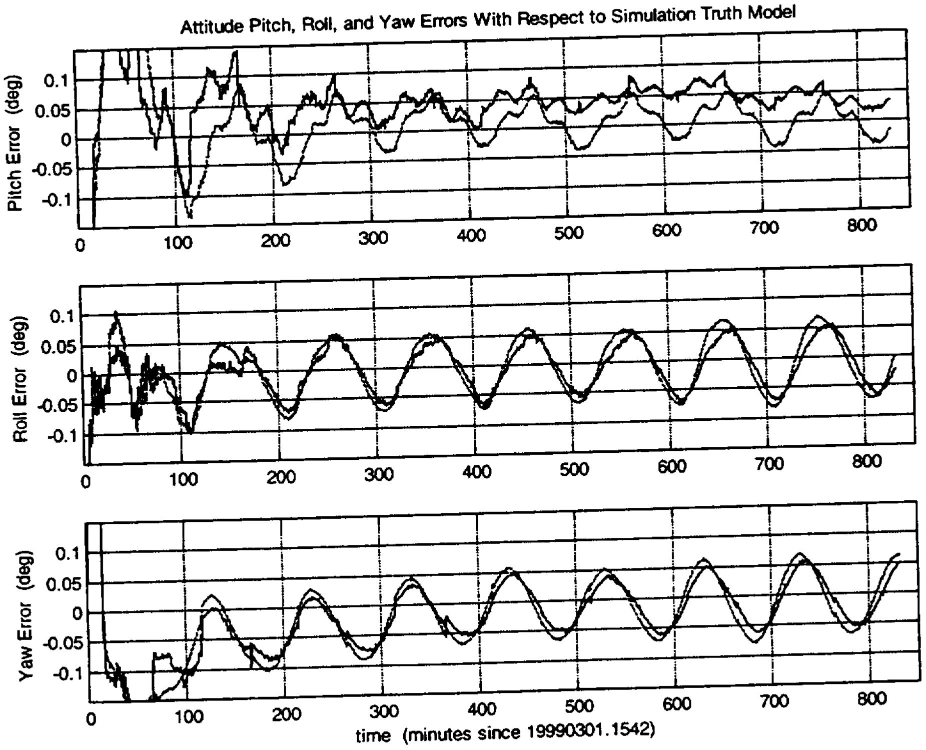


Figure 6. Attitude error with respect to the simulation truth model for the correlated noise filter (solid lines) and for the standard EKF (dashed lines).

apparent that the correlated noise filter fails completely to improve the roll axis (the persistent oscillation of the roll and yaw errors is due to roll/yaw coupling). There is no improvement over the EKF (solid and dashed lines are similar). There is some reduction in the periodic error on the pitch axis, but the mean error is worse. Over many tests, the mean pitch error usually is found to be somewhat worse and its standard deviation better than that of the EKF.

Thus, although the filter does convert the Earth sensor residuals into a nearly white noise sequence, there apparently is a conspiracy of errors between the attitude (used to rotate the reference vector into the body frame) and the estimated systematic error (used to correct the residuals) that leads to attitude estimates nearly unchanged from the EKF. This inability to distinguish errors in separate state components is discussed in Section 4.

Figure 7 shows the results of a test similar to that in Figure 6, however in this case a three-axis magnetometer (TAM) is included as an additional input sensor. The TAM data significantly improve the estimate of the pitch angle. The IRU bias estimates also were improved. However, the roll/yaw errors are still not eliminated.

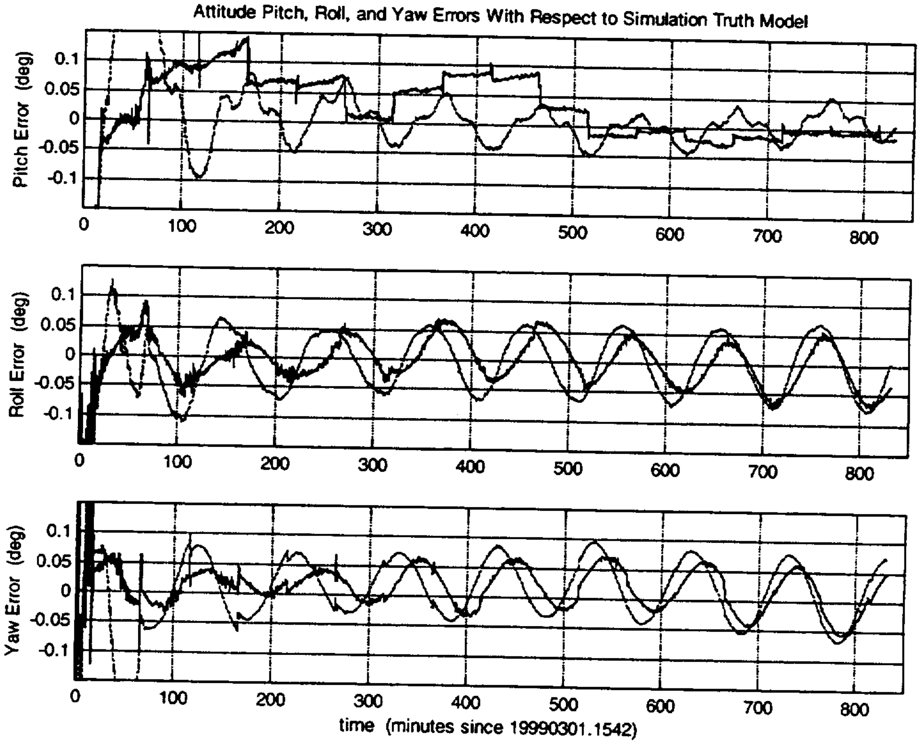


Figure 7. Attitude error with respect to the simulation truth model for the correlated noise filter (solid lines) and for the standard EKF (dashed lines). Input to the filters include data from 1 Earth sensor, 2 DSSs, and 1 TAM.

4. Observability

The tests given in the previous section demonstrate that there is a major problem with estimating Earth sensor errors simultaneously with the attitude and IRU bias. This section presents an analysis showing that the problem derives from the physical system and is not simply an error in the filter equations or implementation.

The observability matrix, Ξ, will be constructed for a deterministic system similar in the essentials to the TOMS-EP scenario. Discussion of this type of analysis can be found, e.g., in Reference 9. The construction requires the observation sensitivity matrix and the state transition matrix.

The error state transition matrix for the attitude, IRU bias, and 2-component Earth sensor correction over a short time interval Δt is approximately

$$\phi = \begin{bmatrix} 1 & 0 & \omega\Delta t & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ -\omega\Delta t & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

where ω is the orbital pitch rate (0.001 rad/sec) about the spacecraft negative y-axis. The correlation time is assumed to be much larger than Δt so the Ψ-block (see Eq. (11)) is near unity.

The sensor measurements for this case can be taken to be a single DSS observation followed by several Earth sensor observations. The sensitivity matrices are given in Eqs. (22) and (23). The body frame directions of the reference vectors are needed in these expressions. For the Earth sensor, the nadir is always near the body z-axis. For the Sun sensor, the Sun becomes visible nearly along the body x-axis as the spacecraft flies over the Pole. This gives

$$H_{ES}^{a} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{25}$$

and

$$H_{DSS}^{a} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{26}$$

The observability matrix (Reference 9) then is

$$\Xi = \begin{bmatrix} H_{DSS}^{a} \\ H_{ES}^{a} \\ H_{ES}^{a}\phi \\ H_{ES}^{a}\phi^{2} \\ H_{ES}^{a}\phi^{3} \\ H_{ES}^{a}\phi^{4} \\ H_{ES}^{a}\phi^{5} \\ H_{ES}^{a}\phi^{6} \end{bmatrix} \tag{27}$$

The system is observable only if the rank of Ξ equals the dimension of the state vector. The rank of Ξ can be determined either numerically or by careful inspection. In this case, the rank is found to be only 7, so the full 8-component system state vector cannot be determined.

5. Conclusions

In an attempt to improve the performance of Earth sensors for spacecraft attitude determination, a simple and well-known technique was used. This involved augmenting the system state with components designed to compensate for systematic errors in the reference model. The noise model was taken as a 1st-order Markov process, which yields an exponentially decaying autocorrelation for the estimated error. However, it was found that this state is not observable given the geometry of the sensors. Inclusion of a magnetometer did not entirely remove the problem. This is surprising because the time-dependence of the observed geomagnetic field over several orbits was expected to give full observability of the state. Further study of this problem is needed.

If the observability question can be resolved, this type of filter could prove to be valuable for use both in ground support and in flight software. There is only a little added complexity in the filter implementation compared to an EKF for attitude and IRU bias. The new filter is only mildly sensitive to the tuning parameters, so robustness and in-flight tuning should not be problems. The main potential benefit would be to eliminate the need for predicting seasonal horizon radiance variations. Rather than using empirical fits to historical data, each spacecraft would estimate its own corrections for its own orbital path and timeframe.

Acknowledgments

The author would like to thank Joseph Hashmall and Dan Andrews for help processing the TOMS-EP data and for helpful suggestions during the analysis.

References

1. M. Phenneger, et al., "The Effects of Seasonal and Latitudinal Earth Infrared Radiance Variations on ERBS Attitude Control," AAS 89-390, *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Stowe, VT, Vol. 71-1, August 1989

2. E. Harvie, O. Filla, and D. Baker, "In-Flight Measurement of the National Oceanic and Atmospheric Administration (NOAA)-10 Static Earth Sensor Error," AAS 93-101, *Proceedings of the AAS/AIAA Spaceflight Mechanics Meeting*, Pasadena, CA, February 1993

3. S. Sallard, et al., "New Accurate Infrared Earth Radiance Model for Pointing Accuracy Improvement of LEO Platforms," AAS 98-337, *Proceedings of the AAS/GSFC International Symposium on Space Flight Dynamics*, Greenbelt, MD, May 1998

4. J. Sedlak and D. Chu, "Kalman Filter Estimation of Attitude and Gyro Bias with the QUEST Observation Model," AAS 93-297, *Proceedings of the AAS/GSFC International Symposium on Space Flight Dynamics*, Greenbelt, MD, April 1993

5. E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman Filtering for Spacecraft Attitude Estimation," *J. Guidance, Control, and Dynamics*, Vol. 5, No. 5, Sept.-Oct. 1982, pp. 417-429

6. J. Sedlak and J. Hashmall, "Accurate Magnetometer/Gyroscope Attitudes Using a Filter With Correlated Sensor Noise," *Proceedings of the NASA/GSFC Flight Mechanics Symposium*, Greenbelt, MD, May 1997

7. J. Sedlak, "Improved Spacecraft Attitude Filter Using a Sequentially Correlated Magnetometer Noise Model," *Proceedings of the 16th AIAA/IEEE Digital Avionics Systems Conference*, Irvine, CA, October 1997

8. A. E. Bryson Jr. and I. J. Henrikson, "Estimation Using Sampled Data Containing Sequentially Correlated Noise," *J. Spacecraft* and *Rockets*, Vol. 5, pp. 662-665, June 1968

9. A. Gelb, ed., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974

MARTIAN AEROCAPTURE TERMINAL POINT GUIDANCE: A REFERENCE PATH OPTIMIZATION STUDY

THEODORE (TED) U. RO[*]
NASA Johnson Space Center, Houston, Texas

ERIC M. QUEEN[†]
NASA Langley Research Center, Hampton, Virginia

SCOTT A. STRIEPE[‡]
NASA Langley Research Center, Hampton, Virginia

ABSTRACT

An effective method of terminal point guidance is to employ influence coefficients, which are solved from a set of differential equations adjoint to the linearized perturbations of the equations of motion about a reference trajectory. Hence, to optimize this type of guidance, one must first optimize the reference trajectory that the guidance is based upon. This study concentrates on various methods to optimize a reference trajectory for a Martian aerocapture maneuver, including a parametric analysis and first order gradient method. Resulting reference trajectories were tested in separate 2000 6-DOF Monte Carlo runs, using the Atmospheric Guidance Algorithm Testbed for the Mars Surveyor Program 2001 (MSP '01) Orbiter. These results were compared to an August 1998 study using the same terminal point control guidance algorithm and simulation testbed. Satisfactory improvements over the 1998 study are amply demonstrated.

INTRODUCTION

The premise of capturing a low lift over drag (L/D) vehicle through a single atmospheric pass (i.e., aerocapture) is arguably the most popular, yet never attempted, method for reducing the amount of on-board propellant necessary to change a satellite's orbit from hyperbolic to elliptic. Aerocapture has been seriously considered in several programs.[§] However, aerocapture has been consistently rejected, partially due to the lack of a guidance routine that meets the required success criteria in Monte Carlo simulations.

A primary reason for the lack of an effective aerocapture guidance routine is the inherent complexity of an aerocapture maneuver. An aerocapture maneuver differs from its landing counterpart in that the degree of complexity is increased due to the target conditions. Specifically, a landing maneuver targets to a specified range (at zero altitude), which is one state variable in the atmospheric re-entry state vector. Conversely, an aerocapture maneuver targets to a *combination* of velocity, flight path angle, and altitude upon atmospheric exit.

The objective of this study is to improve upon a previous study[1] of capturing a low lift over drag (L/D) vehicle into a Martian orbit using "terminal control" to guide the vehicle through an aerocapture maneuver. Terminal control is a guidance method that drives a vehicle to a terminal condition or set of terminal conditions. The purpose of an aerocapture maneuver is to use atmospheric drag to change a satellite's orbit from hyperbolic to elliptic; thus saving propellant that would otherwise have to be transported to Mars. Since aerocapture cannot raise periapsis above the atmosphere, propellant must be

[*] Aerospace Engineer, Ascent/Decent Dynamics Branch, Mail Code DM42, email theodore.u.ro1@jsc.nasa.gov, phone 281 483-6894
[†] Aerospace Engineer, Vehicle Analysis Branch, Mail Stop 365, email e.m.queen@larc.nasa.gov, phone 757 864-6610
[‡] Aerospace Engineer, Vehicle Analysis Branch, Mail Stop 365, email s.a.striepe@larc.nasa.gov, phone 757 864-4512
[§] Orbital Transfer Vehicle studies (1984-85), Aeroassist Flight Experiment (AFE) (1984-89), Mars Sample Return Mission (1986-88), Mars Surveyor Program (MSP) 2001 Orbiter (1997-98), Mars Surveyor 2005 (current)

used to circularize the final orbit. The amount of propellant that is required to place a satellite in a circular orbit is proportional to the required change in velocity or ΔV. ΔV is completely determined by the velocity, flight path angle, and altitude upon atmospheric exit. These exit conditions constitute the aerocapture terminal target. The required ΔV following the aeropass measures the success of the terminal control guidance.

Several Martian aerocapture guidance strategies have been demonstrated in the Atmospheric Guidance Algorithm Testbed[2] for the MSP '01 Orbiter. Tigges et al[3] demonstrated the use of an analytic drag aerocapture controller (ADAC) about a reference trajectory. The ADAC is a derivative of the Aerocapture Flight Experiment (AFE) guidance algorithm. Powell[4] demonstrated the use of a numerical roll reversal predictor-corrector that uses a command vector composed of a roll angle magnitude and times from atmospheric interface to execute roll reversals. Willcockson[5] demonstrated the use of a numerical predictor corrector algorithm that uses bank rate as a control variable for a continuously rolling vehicle. All of the above algorithms are viable candidates for a Martian aerocapture maneuver.

The algorithm described in this paper had the highest success rate (96.1%) in the final 2000 cases 6 degrees-of-freedom (DOF) Monte Carlo simulation of all the guidance algorithms described above. However, this success rate did not meet the 99% success requirement. The difficulty with achieving the MSP '01 success criteria can be attributed to the relatively low available ΔV (130 m/s), the tight inclination range (92.92° ± 0.1°), the vehicle's low lift over drag ratio (0.18), navigational errors, and the dynamic nature of the Martian atmosphere. Partially due to the failure of any algorithm's ability to satisfy the success criteria, aerocapture was deleted from the MSP '01 mission baseline. Given the inherent difficulties associated with the MSP '01 Orbiter's aerocapture maneuver, the authors' believe the MSP '01 testbed represents a substantive environment to further improve their guidance algorithm.

The paper is organized as follows: First, a terminal point guidance algorithm using influence coefficients is described. The strategy is to apply optimal control theory on atmospheric re-entry equations of motion assuming a constant bank angle trajectory to obtain the desired apoapsis. Second, a parametric analysis of entry flight path angle, *constant* bank angle, and exit ΔV is examined to define a reference trajectory. Third, a first-order gradient method is investigated to further optimize a reference trajectory using a *varying* bank angle profile. Fourth, several closed-loop-tuning methods are then discussed. Finally, a plane change maneuver to correct inclination errors is proposed. Numerical results and data are presented where applicable.

SYMBOLS

C_d	=	coefficient of drag
C_L	=	coefficient of lift
D	=	drag, force along the total velocity vector $(C_dS/2)\rho V^2$
f	=	in-plane equations of motion expressed as a function of the in-plane state variables and control vector
g	=	gravity
H	=	angular momentum
h	=	geocentric altitude
h_s	=	scale height
δJ	=	variation in J due to a variation in a control vector for a fixed terminal time
J	=	generic performance index
K	=	overcontrol parameter
L	=	lift, force normal to the total velocity vector $(C_LS/2)\rho V^2$
L/D	=	lift to drag ratio

m	=	mass of vehicle
r_e	=	equatorial radius
S	=	vehicle reference area
s	=	distance
t	=	time
t_f	=	final time of a trajectory
t_0	=	initial time of a trajectory
u	=	generic control parameter
$\delta\Delta V$	=	variation in a reference ΔV due to a variation in a control vector for a fixed terminal time
ΔV	=	difference in velocity required to place a satellite in its desired orbit after an aerocapture maneuver (the performance index used as a target during an aerocapture maneuver)
V	=	total velocity
V_C	=	circular velocity
x	=	in-plane state variables (i.e., the state vector)
$\dot{x}$	=	derivative of state variables with respect to time
δx	=	perturbation of the state vector
ε	=	energy per unit mass
γ	=	flight path angle, positive for climb
λ	=	influence coefficient
μ	=	gravitational parameter for Mars
φ	=	performance index expressed as a function of the in-plane state variables and time
ρ	=	atmospheric density
$*$	=	indicates values taken from the reference trajectory

GENERAL THEORY

The general theory behind terminal control is well documented in Bryson and Ho's *Applied Optimal Control.*[6] Terminal control utilizes a reference trajectory and a set of influence coefficients. These coefficients can be solved from a set of differential equations adjoint to the linearized perturbations of the equations of motion about a reference path.[7]

An optimization problem for dynamic systems involves the following differential equations:

$$\dot{x} = f(x,u,t) \tag{1}$$

$$\dot{\lambda}^T(t) = -\lambda^T(t)\frac{\partial f}{\partial x} \tag{2}$$

where δu(t) is determined by:

$$\delta J = \lambda^T(t_0)\delta x(t_0) + \int_{t_0}^{t_f}\left[\left(\lambda^T\frac{\partial f}{\partial u}\right)\delta u\right]dt \tag{3}$$

and the boundary conditions are:

$$x(t_0)\text{ is given} \tag{4}$$

$$\lambda^T(t_f) = \frac{\partial\varphi(t_f)}{\partial x(t_f)} \tag{5}$$

Note that the particular solutions to equation 1 and 2, respectively, are:

$$\delta x(t) = \frac{1}{\lambda(t)}\left[\lambda^T(t_0)\delta x(t_0) + \int_{t_0}^{t_f}\left[\left(\lambda^T \frac{\partial f}{\partial u}\right)\delta u\right]dt\right] \tag{6}$$

$$\lambda(t) = e^{-\int \frac{\partial f}{\partial x} dt} \tag{7}$$

For a detailed derivation of 6 and 7, refer to reference 1 Appendix A.

From equations 3 and 6, it follows that:

$$\delta J = \lambda^T(t)\delta x(t) \tag{8}$$

Equation (8) provides an important relationship between the effect of a state perturbation on the performance index at any given time.

APPLICATION OF GENERAL THEORY TO AN AEROCAPTURE MANUEVER

The first step in applying the general theory to an aerocapture maneuver is the selection of a performance index (J). The selected performance index was the exit ΔV. This parameter was selected because it is the primary aerocapture requirement metric from the MSP '01 Orbiter project baseline. The exit ΔV is a suitable performance index because it can be defined in terms of the in-plane state variables at a trajectory's final time.

The following matrices are based on in-plane state variables for an atmospheric re-entry vehicle.

$$\mathbf{x}(\mathrm{t}) = \begin{bmatrix} s(\mathrm{t}) \\ V(\mathrm{t}) \\ \gamma(\mathrm{t}) \\ h(\mathrm{t}) \end{bmatrix} \tag{9}$$

$$\mathbf{f}[\mathbf{x}(\mathrm{t}),\mathbf{u}(\mathrm{t})] = \begin{bmatrix} V\cos(\gamma) \\ -\left(\frac{D}{m} + g\sin(\gamma)\right) \\ \frac{L\cos(\phi)}{mV} + \left(\frac{V}{r_e + h} - \frac{g}{V}\right)\cos(\gamma) \\ V\sin(\gamma) \end{bmatrix} \tag{10}$$

$$\lambda(\mathrm{t}) = \begin{bmatrix} \lambda_s(\mathrm{t}) \\ \lambda_V(\mathrm{t}) \\ \lambda_\gamma(\mathrm{t}) \\ \lambda_h(\mathrm{t}) \end{bmatrix} \tag{11}$$

$$\frac{\partial f}{\partial x^*} = \begin{bmatrix} 0 & \cos(\gamma^*) & -V^*\sin(\gamma^*) & 0 \\ 0 & -\frac{2D^*}{V^* m} & -g\cos(\gamma^*) & \frac{D^*/m}{h_s^*} \\ 0 & \frac{L^*\cos(\phi)}{m(V^*)^2} + \left(\frac{1}{r_e + h^*} + \frac{g}{(V^*)^2}\right)\cos(\gamma^*) & -\left(\frac{V^*}{r_e + h^*} - \frac{g}{V^*}\right)\sin(\gamma^*) & -\frac{L^*/m \cos(\phi)}{V^* h_s^*} - \frac{V^*\cos(\gamma^*)}{\left(r_e + h^*\right)^2} \\ 0 & \sin(\gamma^*) & V^*\cos(\gamma^*) & 0 \end{bmatrix} \tag{12}$$

Where the superscript (*) indicates values from a reference trajectory. Further, no angle of attack control was assumed in our analysis.

Substituting the matrices defined above into the equations defined in the general theory derives an optimization equation.

$$\delta\Delta V = \lambda^T \delta x\Big|_{t=t_0} + \int_{t_0}^{t_f}\left[\left(\frac{\partial f^T}{\partial u}\lambda\right)\delta u\right]dt \tag{13}$$

$$\lambda^T(t_f) = \frac{\partial \Delta V(t_f)}{\partial x(t_f)} \tag{14}$$

$$\lambda(t_f) = \left(\frac{\partial \Delta V}{\partial x}\right)_{t=t_f} \tag{15}$$

Since we have chosen the change in our performance index to be zero at atmospheric exit ($\delta\Delta V(t_f) = 0$), we can set the left-hand side of equation 13 to zero and solve for δu. To solve for δu, we must assume δu is a constant.

$$\delta u = \frac{-\lambda^T \delta x\Big|_{t=t_0}}{\int_{t_0}^{t_f}\left(\frac{\partial f^T}{\partial u}\lambda\right)dt} \tag{16}$$

Thus, we have essentially constructed a control parameter (δu) that nulls out the change in the reference $\Delta V(t_f)$.

Next the numerator on the right hand side of the above equation can be re-written with feedback terms. These feedback terms were changed to altitude rate and drag acceleration because they are easily measured quantities. If a continuous determination of the state, x, is made, then a continuous feedback control law results. Further, energy was used as the independent variable because it was determined that velocity had the tendency to converge on a constant value as the vehicle approached atmospheric exit.

$$\delta u = \frac{-\left(\lambda_V(\varepsilon)\delta V(\varepsilon) + \frac{\lambda_\gamma(\varepsilon)}{V\cos\left[\gamma(\varepsilon)\right]}\delta \dot{h}(\varepsilon) - \frac{m h_s(\varepsilon)\lambda_h(\varepsilon)}{D(\varepsilon)}\delta\left[D(\varepsilon)/m\right]\right)}{\int_t^{t_f}\left(\frac{\partial f^T}{\partial u}\lambda\right)dt} \tag{17}$$

Finally, let's evaluate the integral in the denominator on the right hand side of the above equation. One can define this integral as an additional influence coefficient.

$$\lambda_u = \int_t^{t_f} \left(\frac{\partial f^T}{\partial u} \lambda \right) dt \tag{18}$$

The time derivative of this influence coefficient, $\frac{d\lambda_u}{dt}$, is:

$$\frac{d\lambda_u}{dt} = -\lambda_\gamma(t) \frac{D^*(t)}{mV^*(t)} \tag{19}$$

Thus, a control vector that nulls out errors in the performance index at atmospheric exit has been defined in terms of reference trajectory data and easily measurable quantities. Substituting $\frac{L}{D}\cos\phi$ as our control vector results in the following equation.

$$\frac{L}{D}\cos\phi(\varepsilon) = \frac{L}{D}\cos\phi^*(\varepsilon) + \frac{\left(-\lambda_V(\varepsilon)\left(V(\varepsilon) - V^*(\varepsilon)\right)\right)}{\lambda_u} - \frac{\left(\frac{\lambda_\gamma(\varepsilon)}{V\cos[\gamma(\varepsilon)]}\left(\dot{h}(\varepsilon) - \dot{h}^*(\varepsilon)\right) - \frac{mh_s(\varepsilon)\lambda_h(\varepsilon)}{D(\varepsilon)}\left({D(\varepsilon)}/{m} - {D^*(\varepsilon)}/{m} \right) \right)}{\lambda_u} \tag{20}$$

We chose to define our gains as follows:

$$F1 = -\frac{mh_s(\varepsilon)\lambda_h(\varepsilon)}{D(\varepsilon)} \tag{21}$$

$$F2 = \frac{\lambda_\gamma(\varepsilon)}{V\cos[\gamma(\varepsilon)]} \tag{22}$$

$$F3 = \lambda_V(\varepsilon) \tag{23}$$

$$F4 = \lambda_u(\varepsilon) \tag{24}$$

The definition of the above gains results in the following expression for our control law,

$$\frac{L}{D}\cos\phi = \frac{L}{D}\cos\phi^* - \frac{K}{F4}\left\{ F3\left(V(\varepsilon) + V^*(\varepsilon)\right) + F2\left(\dot{h}(\varepsilon) - \dot{h}^*(\varepsilon)\right) + F1\left(\left[{D(\varepsilon)}/{m}\right] - \left[{D^*(\varepsilon)}/{m}\right]\right) \right\} \tag{25}$$

The above equation represents perturbation feedback control or control in the vicinity of a reference path, where K represents an overcontrol factor. The key to this problem is finding an optimal reference path. If an optimum reference path is used, the feedback gains will yield neighboring optimum paths that target to the same terminal conditions as the reference trajectory.

REFERENCE TRAJECTORY OPTIMIZATION THROUGH A PARAMETRIC ANALYSIS

The reference trajectory in the proposed algorithm is critical, because an effective terminal point guidance algorithm that uses influence functions is heavily dependent on an "optimal" reference trajectory. An optimal trajectory is not necessarily the trajectory that results in a minimum or maximum cost function. Certainly, minimizing or maximizing the cost function is important, but when dealing with a closed-loop perturbation feedback control method, one must account for the presence of off-nominal conditions (i.e., dispersions). This reservation is accomplished by analyzing the control vector's profile (e.g., bank angle [ϕ] in our problem). If the reference ϕ profile results in full lift up or down data points, there is a good chance that dispersions will destroy the integrity the guidance algorithm. Hence, for the aerocapture problem, "optimal" is a balance of minimizing the cost function *and* obtaining a robust reference ϕ profile.

Miele and others[8] [9] have shown that if only the atmospheric entry speed is fixed, then the entry flight-path angle can be chosen such that the trajectory that yields a minimal ΔV for post-aerocapture orbital insertion is an one-arc trajectory flown at full negative lift. Evans and Dukeman[10] identified that one must simply determine the shallowest flight-path angle such that, at full negative lift, the vehicle attains the target apoapsis radius at exit. Further, studies have shown that these trajectories are characterized by near-minimal values of peak heating rate and structural loading.[11]

To verify the above, a parametric analysis of entry flight path angle, constant bank angle, and exit ΔV was accomplished within the Atmospheric Guidance Algorithm Testbed. First, optimization runs were performed for various fixed entry flight path angles. The purpose was to identify a range of constant bank angles that resulted in exit ΔV measurements within an acceptable tolerance about the ΔV criteria. With this data, a table of entry flight path angles and constant bank angle profiles was built. A script was used to perform multiple runs utilizing the Atmospheric Guidance Algorithm Testbed. The output file consisted of the entry flight path angle, constant bank angle, and the exit ΔV. A graph of this output file is displayed below.

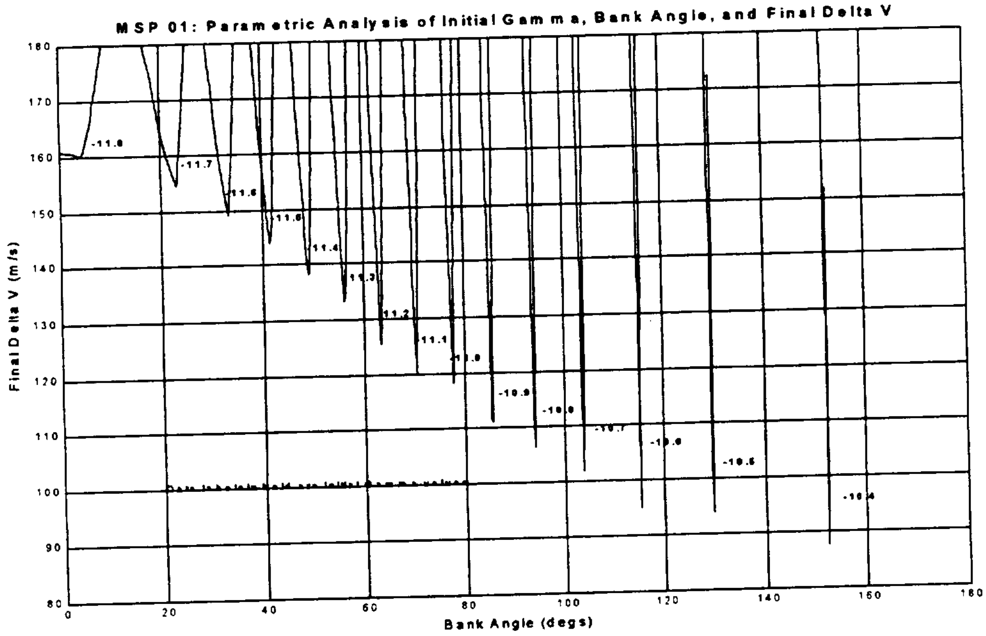


Figure 1 "Parametric Analysis of Initial Flight Path Angle, Bank Angle, and Final Delta Velocity"

Each "spike" represents the bank angle range for a specific entry flight path angle. The entry flight path angles ranged from -11.6° through -10.4°. The exit ΔV range (i.e., the y axis) was limited to the area of interest for the MSP '01 mission. The graph's trend verifies the trajectory that results in the minimum exit ΔV occurs at the shallowest initial flight path angle flown at a 180° bank angle profile. Greater sensitivity to bank angle changes occurs as the entry flight path angle become shallower. Small incremental changes to the entry flight path angle can mean the difference between achieving the desired orbit and skipping out or resulting in extremely large exit ΔV values. The authors were concerned that this sensitivity would manifest itself in closed-loop runs, especially where the nominal entry flight path angle is on the shallower end. In other words, the authors were concerned that entry flight path angle dispersions would have consistent disastrous consequences. However, the results clearly indicate the contrary. An entry flight path angle of -10.36°, which translated to a 176.5° bank angle profile, was used as the shallowest initial flight path angle reference trajectory. From this starting point, several reference trajectories were examined at varying initial flight path angles and bank angle profiles. The closed-loop tuning methods, described later in this paper, were used in all cases. The results from 2000 6-DOF Monte Carlo runs are presented in the next table.

Case #	Trajectory Parameters	Total Actual Success Rate	Total Nav. Success Rate	Actual ΔV Success Rate	Nav. ΔV Success Rate	Avg. ΔV	Actual i Success Rate	Nav. i Success Rate	Avg. i
▨	Rqmt's	≥ 99%	n/a	$\Delta V \leq 130$ m/s			$92.82° \leq i \leq 93.02°$		
▨	1998 Study	96.1%	97.3%	98.67%	98.75%	107.2 m/s	98.2%	99.35%	92.93°
1	γ_f=-10.36° ϕ=176.5°	97.3%	98.45%	98.65%	98.75%	107.2 m/s	98.2%	99.35%	92.93°
2	γ_f=-10.38° ϕ=160.1°	97.05%	98.4%	98.95%	98.9%	105.5 m/s	98.05%	99.45%	92.93°
3	γ_f=-10.40° ϕ=152.3°	97.3%	98.85%	99.3%	99.4%	106.5 m/s	98%	99.4%	92.93°
4	γ_f=-10.45° ϕ=139.0°	95.95%	98.25%	98.9%	99.05%	107.9 m/s	97.05%	99%	92.93°
5	γ_f=-10.5° ϕ=129.2°	94.45%	96.85%	97.4%	97.95%	111.8 m/s	96.85%	98.8%	92.93°

Table 1 "Reference Trajectory Statistical Summary"

The total success rate was based on the formula that a ΔV miss, an inclination miss, or both constituted a failure. Upon first analysis, the total actual success rate increased by approximately 1% from the 1998 study for Cases 1, 2, and 3. The success rate dropped off in Cases 4 and 5, primarily due to the higher nominal exit ΔV in those cases. In all cases, the sensitivity to small incremental bank angle changes in the open loop environment did not manifest itself in closed-loop runs. Given that cases 1 and 3 resulted in the same success rate, further analysis was accomplished on the ΔV distribution for these cases. Case 1 resulted in a higher average ΔV than Case 3. This phenomenon was contrary to the trend of a lower average ΔV as the entry flight path angle became shallower. The ΔV distributions for Cases 1 and 3 are presented below.

	Distribution												
Case #	90≤ ΔV <130	130≤ ΔV <170	170≤ ΔV <210	210≤ ΔV <250	250≤ ΔV <290	290≤ ΔV <330	330≤ ΔV <370	370≤ ΔV <410	410≤ ΔV <450	450≤ ΔV <490	490≤ ΔV <530	530≤ ΔV <570	570≤ ΔV <610
1	1973	19	2	1	1	1	1	0	0	1	0	0	1
3	1986	13	1	0	0	0	0	0	0	0	0	0	0

Table 2 "ΔV Distribution for Cases 1 and 3"

From the above table, it becomes obvious that Case 3 is preferred. Aside from the fact that Case 3 had a higher ΔV success rate than Case 1, the more striking characteristic is the distribution of Case 1. Although Case 1 did not miss the ΔV target often, when it did miss, it missed much worse than Case 3. The effects of a nominal bank angle of 176.5° are clearly displayed in Case 1's ΔV distribution. Case 1 is more susceptible to bank angle saturation because its nominal profile is very close to 180°. If the nominal bank angle is relaxed, as in Case 3's 152.3° profile, the occurrence of extremely high ΔV values disappears. Thus, these results highlight the fact that an "optimal" reference trajectory is a trade between a minimal ΔV value and a robust bank angle profile.

REFERENCE PATH OPTIMIZATION THROUGH A FIRST ORDER GRADIENT METHOD

The previous section examined constant bank angle trajectories. This section investigates the potential for incorporating optimization theory to calculate a *varying* bank angle trajectory. Bryson and Ho[12] discuss numerous iterative procedures to solve nonlinear two-point boundary-value problems, such as in an aerocapture maneuver. The first-order gradient method was selected for analysis because of its extensive use and ability to provide its own convexity as well as the unique combination of known and unknown quantities for this particular problem.

A first-order gradient algorithm for solving the above class of problems is presented below:

STEP (a). Estimate a set of control variable histories, $u(t) = \phi(t)$, and a terminal final time (t_f). Integrate the system equations forward with the specified initial conditions using the control variable history and t_f. Record $x(t)$, $\phi(t)$, $\psi(t_f)=h(t_f)$,

$$\left[\frac{d\Delta V}{dt}\right]_{t=t_f} = \left[p^T f\right]_{t=t_f}, \text{ and} \tag{26}$$

$$\left[\frac{d\psi}{dt}\right]_{t=t_f} = \dot{h}(t_f). \tag{27}$$

As stated above, the reference trajectories utilized in parametric analysis study were put to use. Further, the Atmospheric Guidance Algorithm Testbed which uses the Program to Optimize Simulated Trajectories (POST) as the main integrator, was employed to calculate (i.e., forward integrate) the reference trajectory.

STEP (b). Determine a 4x1 vector of influence functions $p(t)$ and $R(t)$. The vector $p(t)$ represents the influence function associated with minimizing the cost function $[\Delta V(t_f)]$, while $R(t)$ represents the influence function associated with the constraint $[h(t_f)]$. This step is accomplished by backward integration of the following differential equations:

$$\dot{p} = -\left(\frac{\partial f}{\partial x}\right)^T p, \text{ where } p_i(t_f) = \begin{cases} 0 & i = h \\ \left(\frac{\partial \Delta V}{\partial x}\right)_{t=t_f} & i = s,V,\gamma \end{cases} \tag{28}$$

$$\dot{R} = -\left(\frac{\partial f}{\partial x}\right)^T R, \text{ where } R_i(t_f) = \begin{cases} 1 & i = h \\ 0 & i = s,V,\gamma \end{cases} \tag{29}$$

STEP (c). Simultaneously with STEP (b), compute the following integrals:

$$I_{\psi\psi} = \int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u} W^{-1} \left(\frac{\partial f}{\partial u}\right)^T R\,dt, \text{ [scalar]} \tag{30}$$

$$I_{J\psi} = I_{\psi J} = \int_{t_0}^{t_f} p^T \frac{\partial f}{\partial u} W^{-1} \left(\frac{\partial f}{\partial u}\right)^T R\,dt, \text{ [scalar]} \tag{31}$$

$$I_{JJ} = \int_{t_0}^{t_f} p^T \frac{\partial f}{\partial u} W^{-1} \left(\frac{\partial f}{\partial u}\right)^T p\,dt, \text{ [scalar]} \tag{32}$$

where W is an arbitrary positive scalar weighting factor used to help minimize the cost function $[\Delta V(t_f)]$. The choice of W is empirically made based on 1) the discrepancy between the actual $d\Delta V(t_f)$ and its predicted value and 2) the desired step size, where smaller steps are recommended. The predicted value of $d\Delta V(t_f)$ is determined from thc following equation:

$$d\Delta V(t_f)_{predicted} = -\frac{1}{b}\left[\left(p^T f\right)\left(p^T f + \nu\dot{h}\right)\right]_{t=t_f} - (I_{JJ} + I_{J\psi}\nu) \tag{33}$$

STEP (d). Choose a value of $dh(t_f)$ to cause the next reference solution to be closer to the desired value of $h(t_f)$. For example, if the starting $h(t_f)$ does not equal the desired $h(t_f)$, one might choose $dh(t_f)= -\epsilon(h(t_f)-h^f)$, where h^f is the desired value.

STEP (e). Determine ν from the following equation:

$$\nu = -\frac{\left[dh + I_{\psi J} + \frac{1}{b} p^T f \dot{h}\right]_{t=t_f}}{\left[I_{\psi\psi} + \frac{1}{b}\dot{h}^2\right]_{t=t_f}} \tag{34}$$

where b is an arbitrary positive scalar weighting factor used to help drive $h(t_f)= h^f$.

STEP (f). Calculate a new final time by adding dt_f to the original t_f, where:

$$dt_f = -\frac{1}{b}\left(p^T f + \nu \dot{h}\right)_{t=t_f} \tag{35}$$

STEP (g). Calculate a new bank angle (ϕ) profile by adding δϕ to the original bank angle profile. δϕ is determined by the following equation:

$$\delta\phi(t) = -\frac{\left[\left(p(t) + R(t)\nu\right)^T \frac{\partial f}{\partial u}(t)\right]}{W} \tag{36}$$

Use the new t_f value to calculate a time to go percentage. Then multiply the time to go percentage to the original t_f value to properly cross-reference the original reference trajectory data to the new ϕ profile. For example, the time to go percentage can be calculated by the following equation:

$$ttg = \frac{t - t_0}{newt_f - t_0} \tag{37}$$

And the proper cross-reference can be made by the following equation:

$$t_{xref} = ttg(oldt_f - t_0) + t_0 \tag{38}$$

Repeat STEPS (a) through (g) until $h(t_f)$ equals the desired value,

$$\left[\left(\frac{d\Delta V}{dt} + \nu \dot{h}\right)\right]_{t=t_f} = 0, \tag{39}$$

$$I_{JJ} - I_{J\psi} I_{\psi\psi}^{-1} I_{\psi J} = 0, \text{ and} \tag{40}$$

$$H_{uu} > 0 \text{ from } t_0 > t > t_f \tag{41}$$

to the desired degree of accuracy. A MATLAB program was developed to perform STEPS (b) through (g). The following set of graphs is representative of how the first order gradient method modifies a constant bank angle profile.

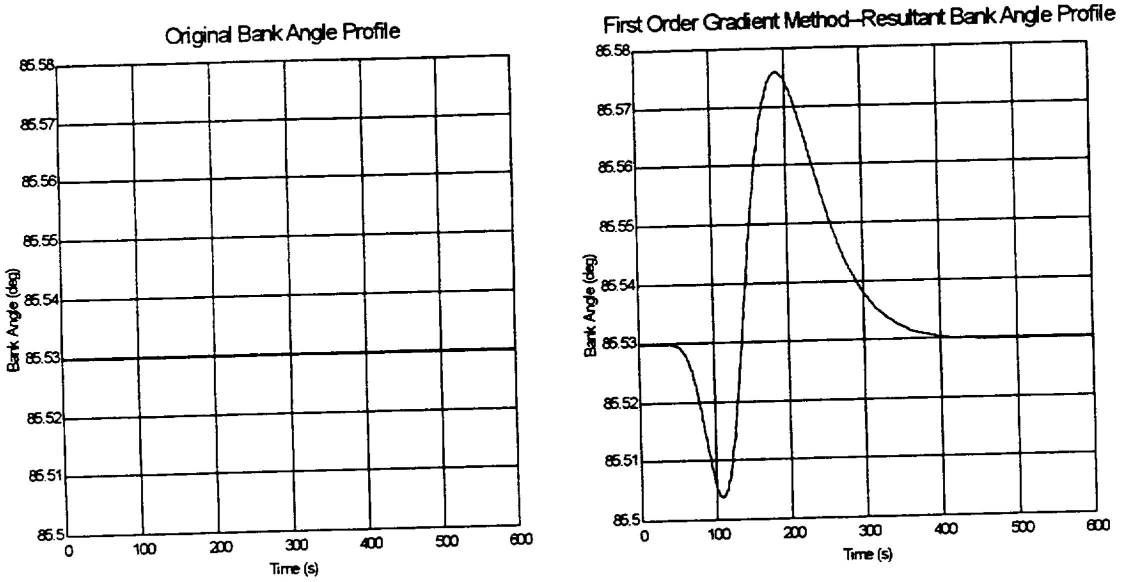


Figure 2 "Comparison of Bank Angle Profile"

Although use of this method did result in further reductions in the exit ΔV, these reductions were not significant (1-2 m/s reductions). It became apparent that first order gradient methods usually show great improvements in the first few iterations, but have poor convergence characteristics as the optimal solution is approached.

Given the above, a second-order gradient method was investigated in hopes of improving convergence. But second-order methods require three conditions: a convexity condition, a normality condition, and a no-conjugate point condition. These conditions must exist throughout the initial trajectory and all subsequent iterations. Meeting all three conditions is not trivial. Several trajectories and optimization schemes were attempted in hopes of meeting these three conditions. These attempts were unsuccessful.[**]

CLOSED-LOOP-TUNING METHODS

Several closed-loop-tuning methods were analyzed with varying results. These included the use of a new control parameter from the study documented in reference 1 as well as a second reference trajectory and bank angle margin profile for high drag acceleration cases.

The original study (see reference 1) utilized $\cos(\phi)$ as its control parameter. Use of this control parameter assumes the reference L/D profile will be the same in all dispersed cases. This assumption, of course, is not correct. Thus, using $\cos(\phi)$ as the control parameter resulted in a less than optimal bank command issued by the guidance algorithm. To correct this situation, $L/D*\cos(\phi)$ was incorporated as the control parameter.

After executing several 2000 6-DOF Monte Carlo cases, it became apparent that certain cases (≈ 15) consistently resulted in an exit ΔV failure. Upon detailed inspection, most of these cases displayed high drag acceleration dispersions. In an attempt to increase the success rate, a second reference trajectory was incorporated. The goal was to measure the difference between the actual drag acceleration and the reference drag acceleration. When this difference exceeded a pre-defined limit for more than "n" events,

[**] The authors would welcome any suggestions on this issue as well as any other aspect of this paper.

a switch over to the second reference trajectory was accomplished. Although this method decreased the magnitude of dispersions, the overall exit ΔV did not make significant improvements. This was primarily due to the fact that a reference trajectory with a higher drag acceleration profile resulted in a higher reference exit ΔV. Maintaining the triggering mechanisms, a further attempt was made to incorporate a second commanded bank angle limit for the higher drag acceleration cases. The commanded bank angle is limited to provide inclination control Nominally, the operational bank angle range was 25° through 155° for a positive roll and -15° through -165° for a negative roll. The asymmetric bank angle limit was a result of the MSP '01 Orbiter's near polar orbit coupled with the planet's oblateness and rotation effects. These effects and near polar orbit resulted in a tendency of the vehicle to naturally drift towards a smaller inclination, which coincided with a negative roll. Thus, less bank angle reserve was required for a negative roll, which provided greater in-plane control. For high drag acceleration cases, inclination control was sacrificed by relaxing the operational bank angle range to 15° through 165° for a positive roll and -8° through 172° for a negative roll. This attempt was based on the logic that higher drag acceleration dispersions require greater in-plane control authority. By "opening" up the commanded bank angle limit, greater in-plane control authority is achieved.

COMBINING THE ΔV AND INCLINATION REQUIREMENTS INTO A SINGLE PERFORMANCE INDEX

The MSP '01 Orbiter aerocapture success criteria consisted of two requirements: 1) an exit $\Delta V \leq 130$ m/s and 2) $92.82° \leq i \leq 93.02°$. An inability to meet the ΔV requirement, inclination requirement, or both resulted in a failure. However, there were numerous cases where the exit ΔV was well under the 130 m/s requirement, but failed to meet the inclination requirement. In such cases, any remaining ΔV theoretically could be used to perform a plane change to meet the inclination requirement. Typically, plane change maneuvers are expensive in terms of ΔV. But the required plane changes are small in our problem. Thus, an analysis on how much ΔV was necessary to correct final inclination errors was accomplished and is documented in this section.

We seek to do a plane change by means of an instantaneous burn at the ascending node. The burn will be perpendicular to the velocity vector and parallel to the local horizon. The figure below depicts this maneuver.

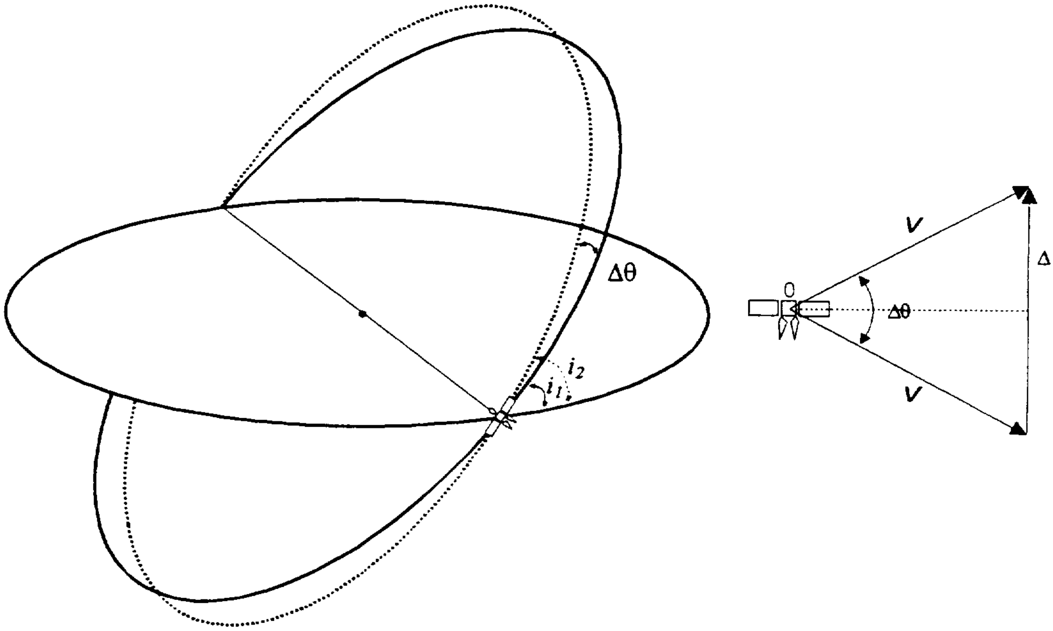


Figure 3 "Plane Change Geometry"

Based on the above figure, the plane change equation is:

$$\Delta V = 2V_c \sin\left(\frac{\Delta\theta}{2}\right) \tag{42}$$

where V_c is the circular velocity.

The above formula combines the previous two mission criteria into one performance index. Based on the way the success criteria were previously defined, both exit ΔV and inclination requirements were of equal importance. However, by combining both requirements into one performance index, the exit ΔV requirement assumes its rightful place as the primary criterion. The exit ΔV should be the primary criterion because an exit ΔV failure cannot be corrected while an inclination failure may be corrected. Thus, an aerocapture guidance algorithm should be designed with greater emphasis to minimize the number of exit ΔV failures and less emphasis on inclination failures.

The following graphs display the required ΔV as a function of plane change for an areodetic altitude of 400km. The graph on the right zooms in on a reasonable available ΔV range (i.e., ΔV that is available after the required circularization burns).

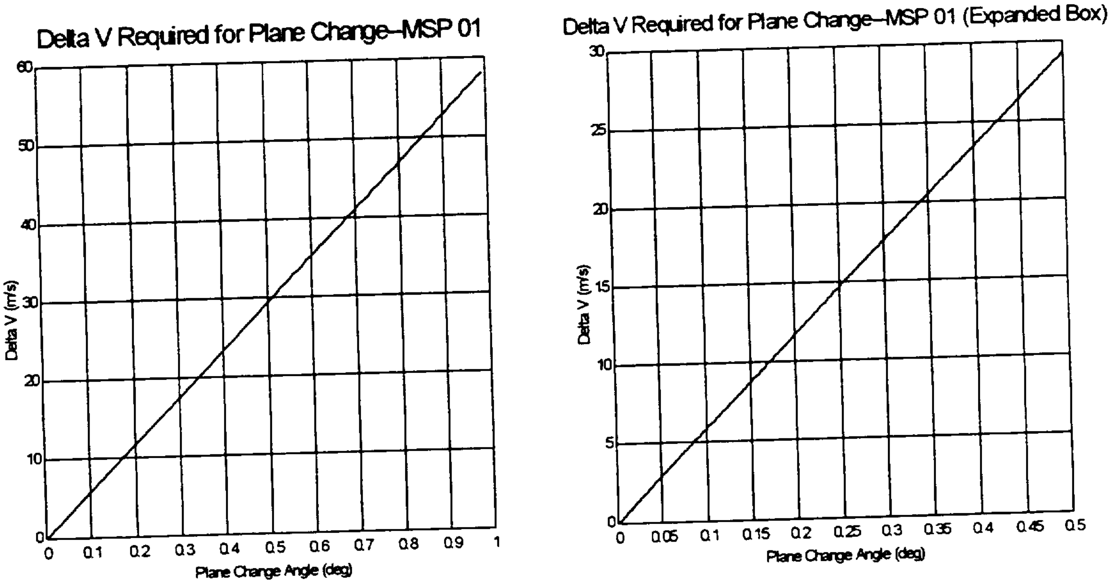


Figure 4 "Delta V Required for Plane Change"

Given that the average ΔV values for the various 2000 6-DOF Monte Carlo runs ranged from 105.5 to 111.8 m/s (ref. Table 1), the reserve ΔV averaged between 18.2 and 25.5 m/s. Thus, based on the graphs in Figure 4, there is on the average at least 0.325° of available plane change ΔV. Applying the above Equation 42 to the 2000 6-DOF Monte Carlo runs results in an updated list of success rates.

Case #	Trajectory Parameters	*New* Total Actual Success Rate	*New* Total Nav. Success Rate	Previous Total Actual Success Rate	Previous Total Nav. Success Rate
	1998 Study	98.6%	98.7%	96.1%	97.3%
1	$\gamma_f=-10.36^\circ$ $\phi=176.5^\circ$	98.65%	98.75%	97.3%	98.45%
2	$\gamma_f=-10.38^\circ$ $\phi=160.1^\circ$	98.9%	98.9%	97.05%	98.4%
3	$\boldsymbol{\gamma_f=-10.40^\circ}$ $\boldsymbol{\phi=152.3^\circ}$	**99.3%**	**99.4%**	**97.3%**	**98.85%**
4	$\gamma_f=-10.45^\circ$ $\phi=139.0^\circ$	98.85%	99%	95.95%	98.25%
5	$\gamma_f=-10.5^\circ$ $\phi=129.2^\circ$	97.2%	97.8%	94.45%	96.85%

Table 3 "Success Rate with Additional Plane Change Maneuver"

The graphs in Figure 5 depict the effect of a plane change maneuver for those cases that failed to meet the inclination requirement. Data from case No. 3 ($\gamma_f=-10.40^\circ$, $\phi=152.3^\circ$) were used for these graphs.

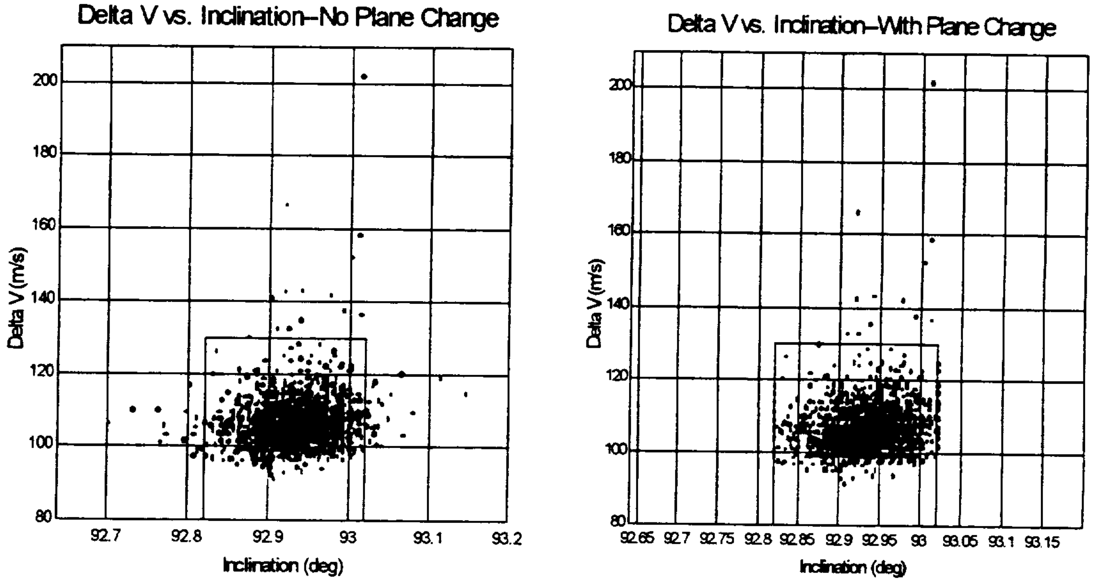


Figure 5 "Delta V / Inclination Box—No Plane Change and With Plane Change"

Notice that in all failed inclination cases, there was enough reserve ΔV (post orbit circularization) to perform a plane change that fixed all inclination errors. The maximum ΔV required to meet the inclination requirements was 7.3101m/s, which translated to approximately a 0.125° inclination error.

CONCLUSIONS

This work improves upon a 1998 study on Martian aerocapture terminal point guidance for the MSP '01 Orbiter. Terminal point guidance utilizing influence coefficients has been proven an effective method for a Martian aerocapture maneuver. This method has displayed excellent performance, even under adverse conditions such as low L/D vehicles, Martian dispersions, relatively low available ΔV, a tight inclination range, and navigational errors. The study of reference trajectory optimization has resulted in acceptable improvements based on the MSP '01 success criteria. A parametric analysis was used to determine the constant bank angle profile that minimizes the post-aerocapture ΔV required to achieve the final orbit conditions. Unfortunately, an attempt to use a first-order gradient method to further optimize the reference trajectory by varying the bank angle profile did not result in satisfactory improvements. The combined use of the reference trajectory optimization work, additional closed-loop tuning methods, and a proposed plane change maneuver to correct for inclination failures increases the success ratio to as high as 99.3% in 2000 6-DOF Monte Carlo runs.

REFERENCES

[1] Ro, T. U., Queen, E. M., "Mars Aerocapture Terminal Point Guidance," Paper No. 98-4571, AIAA Atmospheric Flight Mechanics Conference, Boston, MA, August, 1998.

[2] Striepe, S. A., et al, "An Atmospheric Guidance Algorithm Testbed for the Mars Surveyor Program 2001 Orbiter and Lander," Paper No. 98-4569, AIAA Atmospheric Flight Mechanics Conference, Boston, MA, August 1998.

[3] Bryant, L. E., Tigges, M., and Ives, D. G., "Analytic Drag Control for Precision Landing and Aerocapture," Paper No. 98-4572, AIAA Atmospheric Flight Mechanics Conference, Boston, MA, August 1998.

[4] Powell, R. W., "Numerical Roll Reversal Predictor-Corrector Aerocapture and Precision Landing Guidance Algorithm for the Mars Surveyor Program 2001 Missions," Paper No. 98-4574, AIAA Atmospheric Flight Mechanics Conference, Boston, MA, August 1998.

[5] Willcockson, W. H., "Recent Developments in Aerocapture for the Mars Rover Sample Return Mission," American Astronautical Society, AAS 90-012, February 1990.

[6] Bryson, A. E., Ho, Y. C., Applied Optimal Control, (Hemisphere Publishing Corp., Bristol, PA, 1975), pp. 48-89.

[7] Wingrove, Rodney C., "Survey of Atmospheric Re-Entry Guidance and Control Methods," *AIAA*, J. 9, vol. 1, 1963, pp. 2019-2029.

[8] Miele, A., Basapur, V. K., and Mease, K. D., "Nearly-Grazing Optimal Trajectories for Aeroassisted Orbital Transfer," *Journal of Astronautical Sciences*, Vol. 34, No. 1, 1986, pp. 3-18.

[9] Miele, A., Basapur, V. K., and Lee, W. Y., "Optimal Trajectories for Aeroassisted, Noncoplanar Orbital Transfer," *Acta Astronautica*, Vol. 15, Nos. 6/7, 1987, pp. 399-411.

[10] Evans, S. W., and Dukeman, G. A., "Examination of a Practical Aerobraking Guidance Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3., pp. 471-477.

[11] Miele, A., and Wang, T., "General Solution for the Optimal Trajectories of an AFE-Type Spacecraft," *Aero-Astronautics*, Rept. 254, Rice Univ., Houston, TX, 1991.

[12] Applied Optimal Control, p. 213.

Lunar Prospector Extended Mission

David Folta and Mark Beckman
NASA, Goddard Space Flight Center

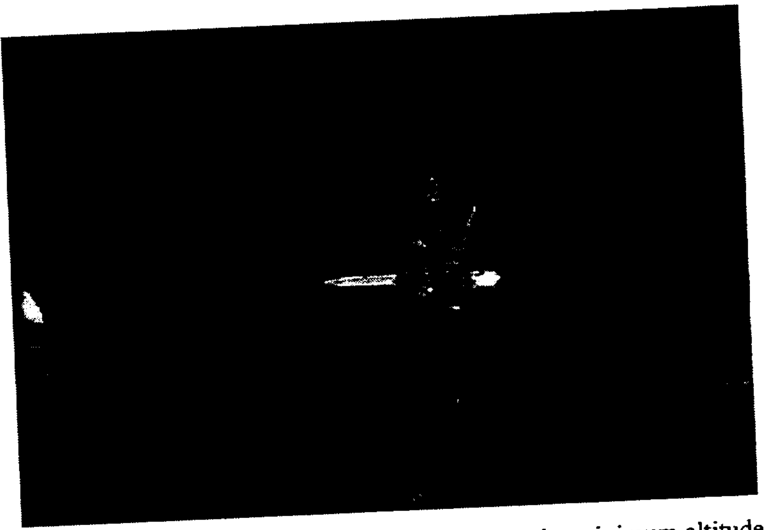
David Lozier and Ken Galal
NASA, Ames Research Center

ABSTRACT

The National Aeronautics and Space Administration (NASA) selected Lunar Prospector as one of the discovery missions to conduct solar system exploration science investigations. The mission is NASA's first lunar voyage to investigate key science objectives since Apollo and was launched in January 1998. In keeping with discovery program requirements to reduce total mission cost and utilize new technology, Lunar Prospector's mission design and control focused on the use of innovative and proven trajectory analysis programs. As part of this effort, the Ames Research Center and the Goddard Space Flight Center have become partners in the Lunar Prospector trajectory team to provide the trajectory analysis, maneuver planning, orbit determination support, and product generation.

At the end of 1998, Lunar Prospector completed its one-year primary mission at 100 km altitude above the lunar surface. On December 19, 1998, Lunar Prospector entered the extended mission phase. Initially the mission orbit was lowered from 100 km to a mean altitude of 40 km. The altitude of Lunar Prospector varied between 25 and 55 km above the mean lunar geode due to lunar potential effects. After one month, the lunar potential model was updated based upon the new tracking data at 40 km. On January 29, 1999, the altitude was lowered again to a mean altitude of 30 km. This altitude varies between 12 and 48 km above the mean lunar geode. Since the minimum altitude is very close to the mean geode, various approaches were employed to get accurate lunar surface elevation including Clementine altimetry and line of sight analysis. Based upon the best available terrain maps, Lunar Prospector will reach altitudes of 8 km above lunar mountains in the southern polar and far side regions. This extended mission phase of six months will enable LP to obtain science data up to 3 orders of magnitude better than at the mission orbit.

This paper details the trajectory design and orbit determination planning and actual results of the Lunar Prospector extended mission including maneuver design, eccentricity & argument of perigee evolution, and lunar potential modeling.



INTRODUCTION

The launch of the Lunar Prospector (LP) spacecraft marked the return of America's space program to the moon, picking up where the Apollo program left off with a low altitude polar orbiting mission to map the entire surface of the moon. Lunar Prospector is a modest spacecraft funded at a cost of $63 million (including the launch vehicle) by NASA's Discovery Program. Six science experiments were flown to map the composition of the lunar surface, study the moon's gravity and magnetic fields, investigate levels of tectonic and volcanic activity, and search for evidence of water ice at the lunar poles. The Lunar Prospector spacecraft is a spin-stabilized graphite-epoxy drum, 1.4 meters in diameter by 1.22 meters in height, with three radial instrument booms located 120 degrees apart. Attitude, spin rate, and velocity control are provided by a blowdown monopropellant hydrazine propulsion subsystem using six 22 N thrusters. Telemetry and command functions are provided by a single S-band transponder through either a medium gain or an omni-directional low gain antenna mounted on a mast aligned with the spacecraft spin axis. The total spacecraft mass at launch was 296.4 kg, including 137.7 kg of hydrazine propellant. At the beginning of the extended mission, 17.41kg of fuel, 12.61%, was available.

Extended Mission Profile

A description of the nominal mission profile for the Lunar Prospector mission has been previously presented (ref 1 and 2). The Goddard Space Flight Center's Guidance, Navigation, and Control center (GN&CC) and the AMES Space Projects Division were tasked to provide mission design, maneuver support and orbit determination. These data and related products such as latitude and longitude, occultation, and station visibility can be found on the GN&CC LP Product Center web site at *http://fdd.gsfc.nasa.gov/lp* .

The goal of the six month LP extended mission is to collect low altitude instrument measurements that provide higher resolution data (particularly from the magnetometer and Neutron spectrometer) to complement measurements collected during the nominal mission. The extended mission design goal was to place the spacecraft in a mean 25km altitude orbit while restricting altitude excursions between a minimum of 10 km and a maximum of 40km. This mean altitude was later changed to 30km as discussed herein. As part of the extended mission analysis, a transition orbit was used. This transition orbit had a mean 40km altitude. The transition orbit allowed verification of the operations plans and gravitational modeling before executing the final maneuvers to meet the extended mission altitude goals.

LUNAR ORBIT EVOLUTION AND MISSION DESIGN

While a similar strategy is used for maintaining orbital eccentricity and argument of periapsis as that used for the primary mission, the lower altitude associated with the extended mission reduces the range of allowable eccentricity growth, and necessitates shorter time periods between maneuvers. Furthermore, the lower altitude increases the sensitivity of the orbit to perturbing gravitational forces and increases the influence of initial orbit conditions on orbit evolution. While the altitude is much lower, the orbit mean sma only changed from 1838 km to 1763 km, a 4% decrease. The primary mission orbital altitude was subject to a +/-20 km science constraint about the mean 100 km altitude and varied on this order (+/-15 km). The argument of Periapsis (ω) and eccentricity (e) also evolved in a predictable pattern. This evolution pattern was one of the first astrodynamic verifications of the primary mission. The variation in both e and ω is a direct result of the lunar potential and requires the latest potential and the ability to model these accelerations. Frozen orbit e and ω conditions of ~0.04 and 270 degrees result in either a lunar impact while trying to meet the required science goal of a 30 km mean circular altitude or drive the apoapsis higher than allowed while maintaining a low periapsis. Therefore, a Quasi-Frozen Lunar Orbit (QFLO) method is being used (ref 3).

Figures 1 shows a phase space plot of the evolution of the primary 100 km orbit with e the radial component and ω the angular component. While the figure shows a predictable evolution, the constraints of the extended mission require a watchful eye on the selenographic altitude and orbit eccentricity. A predictable orbit evolution is made possible by the correct usage of high fidelity modeling of the lunar potential, the Earth and sun gravity, solar radiation pressure accelerations, and maneuver perturbations. For the extended mission a 100 degree and order lunar potential model was utilized for orbit determination, orbital predictions, and maneuver planning. A 75 degree and order had been used for the primary mission.

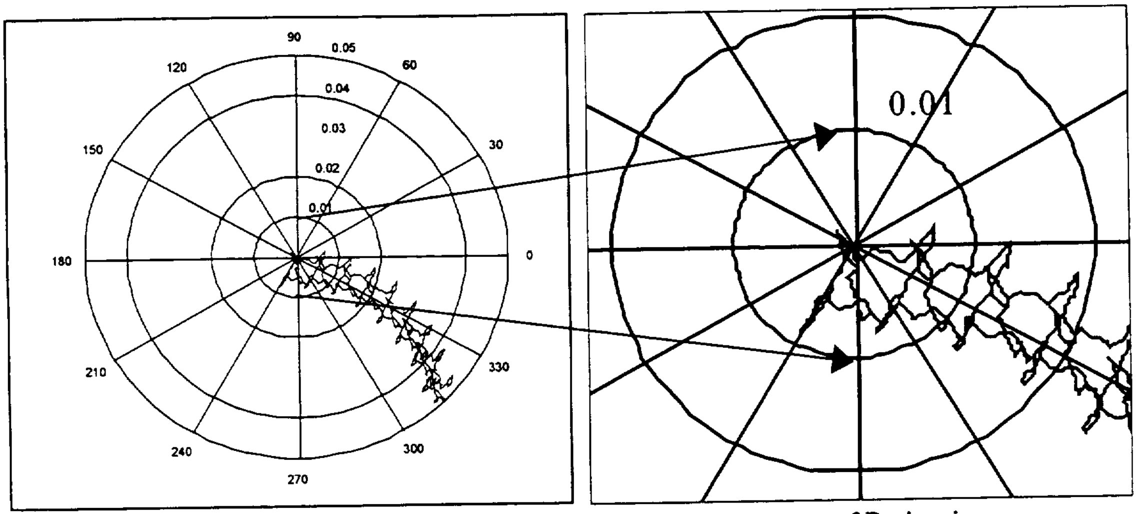


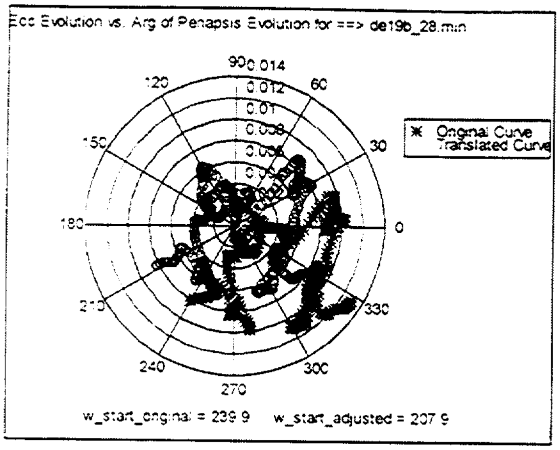
Figure 1: Phase Space Plot of Eccentricity vs. Argument of Periapsis

To help understand the evolution, a simple geometric method is devised to adjust the orbit evolution curve in phase space coordinates in order to minimize excursions in eccentricity, and thereby arrive at an optimum argument of periapsis starting condition. Figure 2 shows an example of a December 19, 1998 55-day transition orbit (40 km +/-15 km initial altitude) with an original ω value of 240 degrees, which has been adjusted to yield a new suggested ω of 208 degrees. While the original curve failed as a candidate orbit to meet minimum altitude requirements (20 km in the transition orbit), the new translated curve succeeds by minimizing eccentricity (and consequently altitude excursions) over the 55-day duration. This geometric method is approximate in that it ignores small changes to the shape of the evolution curve as a result of different starting values of ω, nevertheless, it has been shown to produce fairly accurate results. As such, the geometric method is a useful tool to obtain quick convergence on the required initial conditions of ω when designing extended mission orbits.

Note the ω vs. e pattern in the geometric figures. This pattern is repeated at approximately the same order of magnitude with the pattern collapsing on itself in the ω term for the quasi-frozen orbit (ref 3). The initial conditions and the selection of the epoch determine the starting location on the pattern in the figure.

Epoch Dependency

While the figures presented thus far depend on the initial argument of periapsis and eccentricity, the selection of the epoch for these conditions is also important. The selection of a different epoch results in the start somewhere else on the pattern, but does not change the shape of the pattern evolution. This is shown in Figure 3. The two epochs differ by 14 days, half of the pattern cycle time. While the initial conditions are the same, the shape did not change. The change in epoch moved the starting point of the pattern to the middle for this case. In Figure 3, note that the right half of the pattern labeled A is the left half of the pattern labeled B even though they have the same initial orbital conditions.



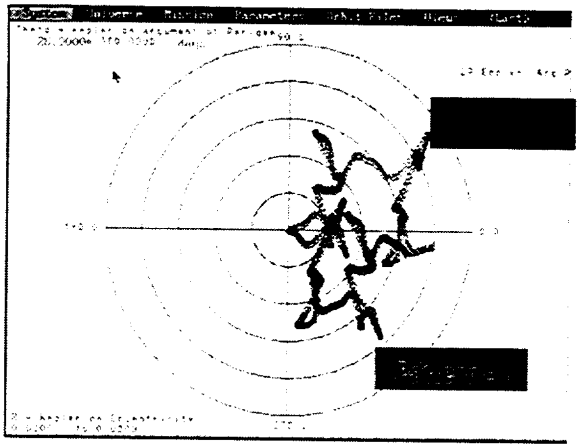


Figure 2: Post December 19 Maneuver Evolution

Figure 3: Evolution Dependency on Epoch

The Selection of the Extended Mission Mean Altitude

The original goal of the extended mission was to have a mean altitude of 25 km with a tolerance of 15 km. As we began design of this orbit, we incorporated the latest databases on selenographic altitude of the surface. We quickly realized that the altitude above the estimated surface features for a 25 km mean altitude could place us at a periapsis selenographic altitude of less than 4 km.

In order to predict the actual altitude over the varying lunar terrain, an accurate topography model was needed. The final topography model used was combined from two different sources. Low latitude data was obtained from models developed from the 1994 Clementine mission's LIDAR data. High latitude data was obtained from a Clementine stereo imaging Digital Elevation Model (DEM).

The processed DEM model gave maximum altitudes over each one degree by one degree bin with noise suppression which excluded any elevation spikes over 9.6 km. The data was also interpolated over regions where no data was available. The processed LIDAR model gave averaged altitudes over each 1x1 degree bin. Maximum altitudes were not available. The combined model gave altitudes over a 1x1 degree grid of the lunar surface. The actual trajectory spherical altitude was compared to the nearest topography model altitude to obtain a best estimate of actual altitude.

The combined model used the LIDAR data from ± 59 degrees latitude and the stereo data at greater absolute latitudes. The two models admittedly did not merge into a smoothed database; there are discontinuities of up to 6 km at the merged latitudes. Figure 4 shows a contour map of the combined database. The differences in the generation of the models used and the data processing into 1x1 degree bins prevents a more accurate database. Because of the uncertainties in this final model, it was decided to lower LP to only a 30 km mean altitude orbit which will place the periselene no lower than 12 km and not near any of the known high altitude regions.

Two Week versus Four Week Evolution

The decision to maintain an altitude of 30 km vice the 25 km was really a reflection of several factors: the combined effect of the predicted e and ω evolution, the anticipated selenographic surface altitude, the selection of targets for maneuver planning, more frequent operations, fuel considerations, and occultation free periods. At the 25 km altitude, the eccentricity evolution does not allow maneuver free durations of greater than two weeks, but would allow occultation free periods for the execution of the maneuvers. This more frequent maneuver schedule would also drive the operations schedule for GSFC maneuver, orbit determination and planning products. During the primary mission orbit maneuvers occurred every 56 days, whereas the extended mission maneuvers would need to occur every 14 days. Also the maneuver ω target would alternate every 14 days while the periapsis and apoapsis targets would be the same for either case since the goal is to maximize the time between maneuvers. Figures 5 and 6 show an example of maintaining a 25 km altitude every two weeks.

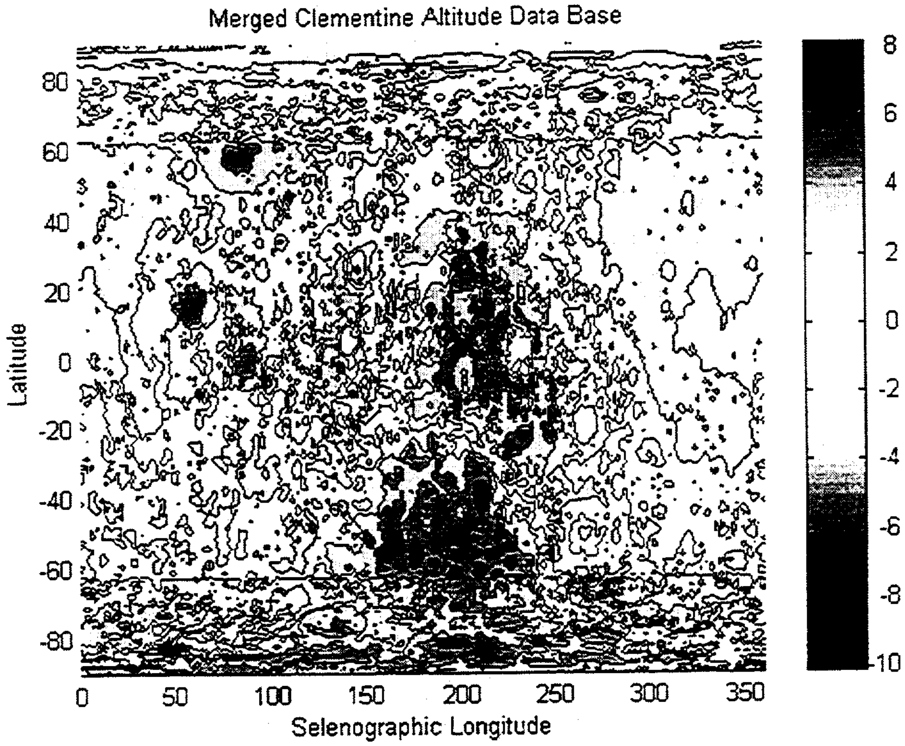


FIGURE 4: Merged LIDAR and Stereo Imaging Lunar Topography Map

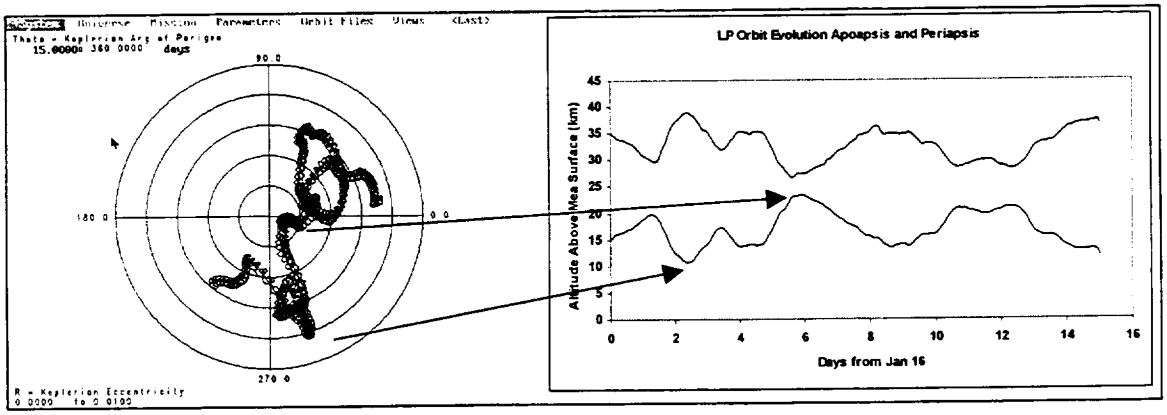


Figure 5: Two Week Evolution And Altitude Variation, First Two Weeks

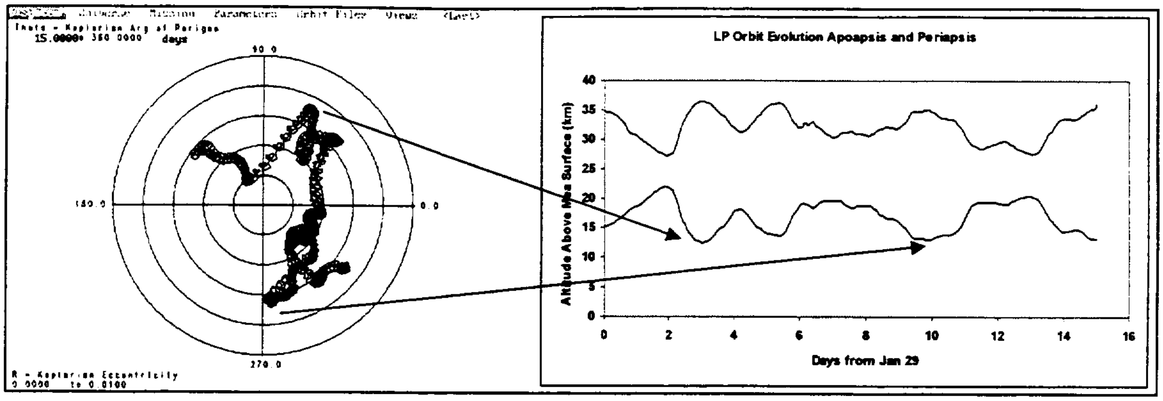


Figure 6: Two Week Evolution And Altitude Variation, Second Two Weeks

Tables 1 and 2 present maneuver information for the two and four week scenarios. The ω targets are contingent upon the evolution sequence. The table shows the maneuver dates during our planning effort before the extended mission. The major differences in the scenarios are an increase in the ΔV magnitude approximately 20%-40% for the four week maneuvers but a decrease in maneuver frequency. The four week scenario uses the same ω and e targets for all maneuvers. Table 2 provides the changes in the orbital parameters corresponding to a decrease in the sma for the extended mission. The major impacts of the lower sma are an increase in the lunar eclipse duration, which has an effect on the LP power conditions. This condition requires an on/off cycling of the transmitter to allow for battery recharge upon exit from the shadow. Another major impact is the reduction in half of the occultation free period. Since we are required to perform maneuvers during visibility, the maneuvers must be executed on a particular date. The result of missing a maneuver by two or three days is impact with the lunar surface.

Table 1. Two and Four Week Maneuver Scenario

Orbit Type		sma (km)	ΔV1 (m/s)	ΔV2 (m/s)	ω (deg.)
Two Week Scenario					
12/19/98	Transition	1778 (28 day)	-21.4	6.33	200
1/16/99	EMOC#1	1763 (14 day)	-1.2	-7.8	230
1/29/99	EMOC#2	1763 (14 day)	-5.1	-4.0	145
2/12/99	EMOC#3	1763 (14 day)	-3.9	-3.9	230
2/25/99	EMOC#4	1763 (14 day)	-4.0	-4.0	145
Four Week Scenario					
12/19/98	Transition	1778 (28 day)	-21.4	6.33	200
1/16/99	EMOC#1	1768 (28 day)	-4.0	-7.6	205
2/12/99	EMOC#2	1768 (28 day)	-7.5	-5.5	205

Table 2. Extended Mission Orbital Parameters

Circular Mapping Orbit Altitude	Orbit Period (min.)	Max Ecl./Occ. (min.)	Min Daylight (min.)	Occ. Free Period (hrs.)	Eclipse Season (dates)
100 km	117.85	46.49	71.36	48-72	12/16/98 Ends
40 km	112.13	48.48	63.65	30-40	12/19/98 to 12/24/98 Ends
25 km	110.71	49.41	61.30	24-36	1/14/99 to 6/25/99 Ends 7/17/99 Start

Selenographic Location of Periapsis

During the planning of the maneuver from the transition orbit to the extended orbit, it was noted that an extension of two weeks in the 40 km transition orbit would result in the periapsis selenographic latitude and longitude being rotated to the Earth side of the lunar surface. This would allow new science data collection over regions at yet unexplored lower periapsis altitudes. Additionally, once in a 30 km altitude orbit, the periapsis would not rotate to the Earth side before a maintenance maneuver would be required. It was decided to delay the maneuver to lower the orbit to the final extended mission by two weeks. This also deferred all the subsequent maneuvers by two weeks until May when we will change the periapsis location back to the far side.

EXTENDED MISSION MANEUVER PLANNING

The strategy for maintaining the extended mission polar orbit was developed with the following goals:

- Maintain an altitude band of 30 km +/- 15 km, with a minimum altitude of 10 km
- Conduct maneuvers in view of a ground station
- Minimize the number of maneuvers
- Use axial maneuvers instead of vector (axial and tangential) burns if possible

The last goal was established for reasons of operational simplicity, since LP vector burns cannot be performed readily during shadow periods for lack of a reference sun pulse. The nominal LP spin-axis attitude is within a few degrees of the ecliptic normal and therefore almost normal to the lunar equator. To meet this goal requires that the argument of periapsis be close to zero or 180 degrees to allow axial maneuvers to take place parallel to the velocity direction at periapsis / apoapsis. Furthermore, as LP maneuvers consist of a two-burn Hohmann like sequence, the second goal requires that maneuvers be conducted when the orbit plane is normal to the Earth/moon line, a condition that occurs every 13.7 days. A one burn strategy was considered for contingencies and for minimizing operations, but it would result in a considerably less efficient vector burn. Also, a one-burn placed at the poles did not always allow the ω requirement to be met. Shown in Figure 7 is a representation of the locations and the orientation of the orbit plane for all extended mission maneuvers. The maneuvers are performed first at the apoapsis and then near the ascending node to both meet requirements and to provide the most efficient locations.

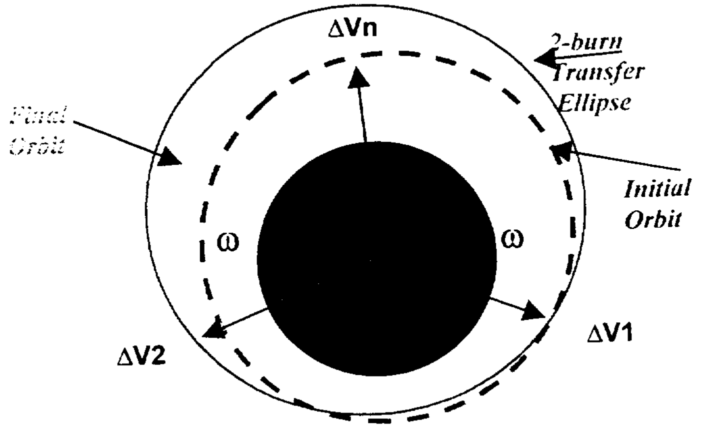


Figure 7: Maneuver Placement

EMOC Numerical Targeting Process

For extended mission targeting, a numerical method to propagate and to achieve target goals is required. While the geometric method is based on a pattern recognition, which minimizes the exclusion beyond the minimum altitude goal based on the e amplitude, it is not optimized. To support the LP mission design concepts presented in this paper, GN&CC operational software called Swingby is used (ref 4). This software provides a highly accurate numerical generation of data, which allows the inclusion of the latest lunar potential models, third body perturbations, and Solar Lunar and Planetary files. It has targeting capabilities, which include a Differential Corrector (DC) and two optimizers; a steepest decent method and a Quasi-Newton (QN) method. This software also provides for the computations of the maneuvers to maintain the orbits and the generation of the finite burn based on the spacecraft propulsion and attitude properties. The numerical method of choice consists of a Runge-Kutta Nystrom or Verner 8/9 variable step propagator, a differential correction process to target the periapsis and apoapsis altitude and ω, and the use of the spacecraft attitude, mass properties, and blowdown coefficients in the finite burn modeling.

The maneuver planning process is as follows:

- Update initial state from definitive solution
- Propagate to maneuver condition of occultation free zone
- Propagate to apoapsis or periapsis condition or node crossing condition
- Maneuver using axial thrusters: impulsively and/or finite burn model
- Target using a differential corrector to meeting goals of e & ω and propagate to ensure next maneuver conditions and correct altitude evolution
- Initial Δv guess based on experience, previous maneuver, and impulsive computations

EXTENDED MISSION RESULTS

Transition Orbit

Figure 8 shows the Swingby output from the transition maneuver planning sequence. The plots in the figure show the polar phase space plot of e vs. ω, the spherical periapsis altitude above the mean lunar geode, the selenographic periapsis altitude based on the merged surface database, and the selenographic latitude and longitude. The evolution plot of e versus ω is centered on zero eccentricity with ω increasing counterclockwise. The maximum eccentricity represents the lunar spherical surface. The latitude and longitude are oriented so that the center of the plot represents the far side of the moon. The drift of e and ω

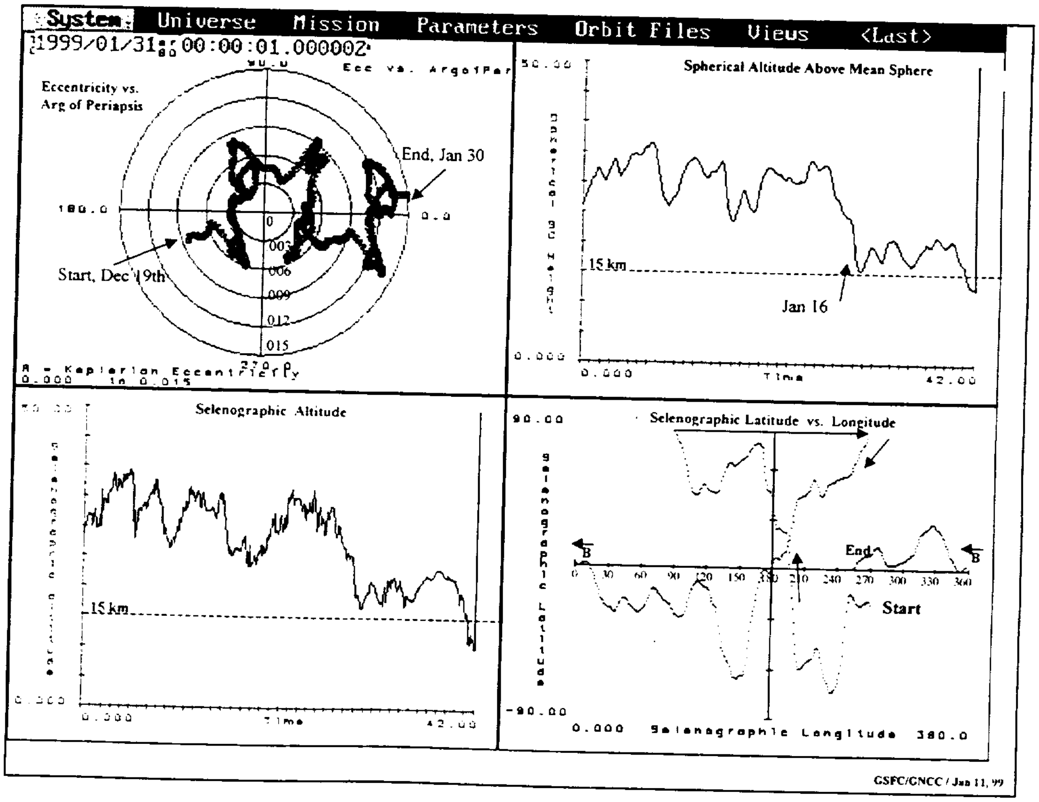


Figure 8: Swingby Output from the December Maneuver Plan

proceed to the right on the phase space plot. The right side of the phase space plot shows the extension of the two week 'wait' for the flip of the latitude / longitude to the Earth side. The effect of the two week delay is also seen in the extended mission altitude plots on the right side where the altitude is significantly lower due to the eccentricity growth. At the end of January, the first EMOC maneuver took place.

Extended Mission Orbit Control (EMOC)

Figures 9 and 10 show a time history of the LP phase space and the orbit periapsis/apoapsis altitude through the first two EMOCs. To date, the availability of an accurate lunar potential model has been a key factor in the successful planning of LP mapping orbit maintenance maneuvers. Using the post maneuver definitive states, a prediction of the orbit evolution is plotted. In order to limit excursions beyond the +/- 15 km range, initial values of 155 deg for argument of periapsis and approximately 0.0085 for eccentricity (15 x 45 km orbit) were targeted as maneuver end conditions for subsequent maneuvers. This would force the post maneuver orbit to evolve towards the zero eccentricity point and permitted a longer time between maneuvers. As seen in the phase space plot, the pattern of the evolution has been shifted to the right with the eccentricity passing close to zero. This is a direct result of waiting the extra two-weeks for the latitude and longitude drift to occur. Other that that effect, the evolution occurs as predicted. The spherical periapsis altitude in the figures are plotted along with the surface height above the mean lunar geode. The dates of the maneuvers are ~28 days apart and coincident with periods of full station coverage near full moon (i.e. face-on orbit geometry). Furthermore, as the phase space plot shows, the maneuver dates occur when the line of apsides is within 20 degrees of the lunar equator, allowing axial maneuvers to take place with only minor losses in efficiency. Also note the flip in the line of apsides occurring with each maneuver. Without maneuvers, the orbit could be expected to impact the moon within only 2 days with the ω evolving clockwise towards 270 degrees. In Figure 10, the results of the EMOC#2 maneuver are plotted. Note on both altitude plots, the definitive states are plotted as squares to show that the prediction holds true. The difference in the definitive and predictive altitude is only approximately a few tens of meters. All maneuvers were planned as Moon centered in a lunar equator , Mean J2000 coordinate system. Also, EMOCs #1 and #2 were performed using only the axial thrusters and the QFLO method (ref 3).

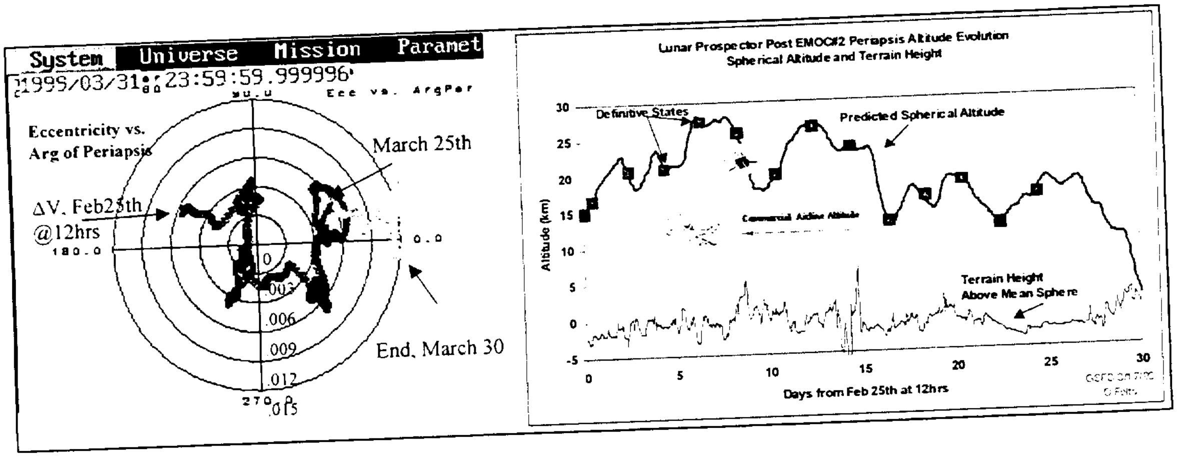


Figure 9: EMOC#1 Results

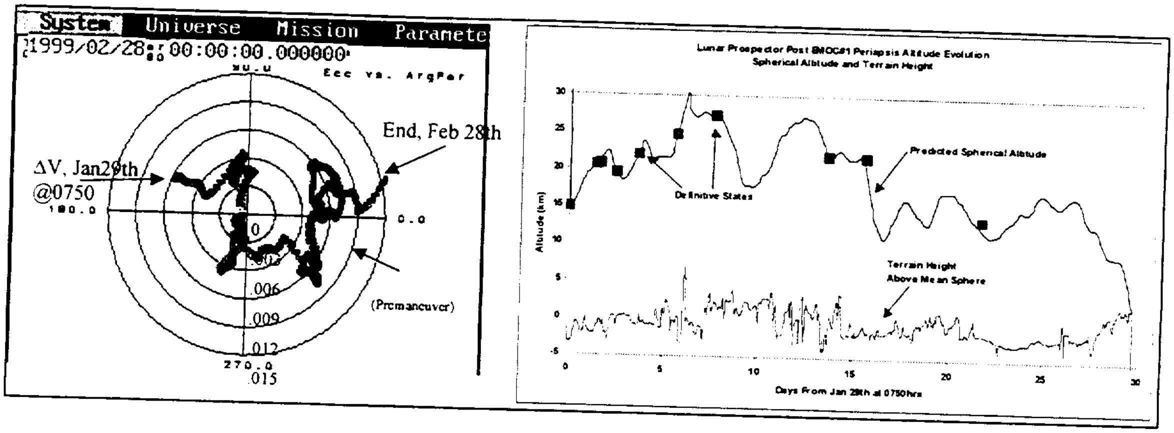


Figure 10: EMOC#2 Results

Thruster Performance

The performance of Lunar Prospector has been excellent as of the end of EMOC#2. Table 3 provides both Mission Operations Correction (MOC) and Extended Mission Operations Corrections (EMOC) post maneuver data. As shown by the estimate performance row, maneuver performance has always been below a repeatability factor of 2%. All post-maneuver states are represented in terms of Mean-of-J2000 Keplerian elements relative to the lunar equatorial plane. The maneuver ΔV magnitudes have been roughly the same for all maneuvers since the change to the eccentricity and sma have also been similar. The states represent the definitive solution shortly after the maneuver. These states were generated using approximately 8-24 hours of Doppler and range tracking data. A short duration was used so that the effects of the potential did not influence the calibration. The calibration was performed by using the Differential Corrector in Swingby to adjust the thrust scale factor parameter while targeting on the periapsis and apoapsis radius at an epoch shortly after the second maneuver.

Table 3 Lunar Prospector Maneuver Summary

	MOC#1	MOC#2	MOC#3	MOC#4	MOC#5	MOC#6	Transition	EMOC#1	EMOC#2
Target Maneuver ΔV	12.1 m/s 2.6 m/s	7.4 m/s 7.3 m/s	6.0 m/s 5.7 m/s	6.1 m/s 6.0 m/s	6.4 m/s 6.8 m/s	7.2 m/s 6.5 m/s	7.2 m/s 6.5 m/s	6.3 m/s 11.3 m/s	6.9 m/s 7.4 m/s
Thrust Scale Factor Assumed	.967	1.0	.944	.944	.925	.967	.967	.967	.967
Estimated Performance	100.4%	94.4%	99%	98%	101.1%	98%	99%	100%	99.6%
Post-ΔV State Epoch (GMT):	1/16/98 00:00:00	3/08/98 05:00:00	5/01/98 17:00:00	6/25/98 23:23:00	8/17/98 17:45:00	10/12/98 19:20:00	12/19/98 19:15:00	01/29/99 08:35:06	02/25/99 12:53:10
a (km):	1838.3	1838.0	1838.0	1837.4	1836.1	1837.8	1778.9	1767.9	1768.0
e :	0.00046	0.00683	0.00645	0.00512	0.00937	0.00795	0.00841	0.0085	0.00843
i (deg):	90.55	89.93	89.88	89.82	89.89	89.75	89.71	89.60	89.67
Ω' (deg):	192.76	192.43	192.52	192.55	193.82	193.83	193.83	193.53	193.39
ω (deg):	224.02	196.26	191.51	195.78	180.71	174.05	200.98	154.99	154.85
MA (deg):	317.04	181.80	184.08	274.88	314.77	203.98	203.98	0.00	0.00
Period (hrs):	1.965	1.964	1.964	1.963	1.961	1.964	1.870	1.853	1.853
$R_{crossover}$ (km):	1839.2	1850.5	1849.9	1846.9	1824.05	1851.2	1793.9	1783.0	1782.9
Central Body	Moon	Moon	Moon	Moon	Moon	Moon	Moon	Moon	Moon

Notes: The MOC#2 through MOC#5 maneuvers used thruster combination A3/A4 for the first time
The MOC#6 through EMOC#2 maneuvers used thruster combination A1/A2 after the 180 degree spin axis attitude flip

ORBIT DETERMINATION

LP's primary mission was flown at a 100 km mean altitude for one year. During the nominal mission, the post-processed definitive ephemeris requirement was 1 km 1-sigma position accuracy in each of radial, crosstrack, and alongtrack. The lunar potential model was the leading source of orbit estimation error. A few months into the nominal mission, Dr. Alex Konopliv at JPL using the most recent LP Doppler measurements, updated the lunar potential model (ref 5). This model, LP75D, was a 75th degree and order model that was considerably more accurate than any pre-LP models. Definitive orbit ephemerides were generated throughout the nominal mission using a batch weighted least squares algorithm. The solution arcs were 55 hrs with a 7 hr overlap between consecutive solution arcs. The definitive orbit accuracy was measured by comparing the ephemerides over this overlap period. This technique is more accurately a consistency measurement, but without independent tracking of the spacecraft, it is the best available technique. The definitive solution accuracy during the nominal mission phase is shown in Table 4 (ref 6). The Root Mean Square (RMS) of the position component differences, measured every 10 min during the overlap period, is considered the 1-sigma position component accuracy. All mapping orbit solutions were performed using Doppler measurements only.

TABLE 4: NOMINAL MISSION ORBIT DETERMINATION ACCURACY

Model	Radial RMS	Crosstrack RMS	Alongtrack RMS	Position RMS	Avg. Doppler Residual
LP75D	13 m	155 m	189 m	270 m	9.3 mm/sec

The definitive solutions, using LP75D, exceeded the mission requirements by more than a factor of five. The average fit to the Doppler measurements was 9.3 mm/sec. A graph of a sample solution arc's residuals is shown in Figure 11. A magnification of the graph is shown in Figure 12. The residual signature is clearly not due to noise but exhibits a distinct pattern, most likely due to potential model errors.

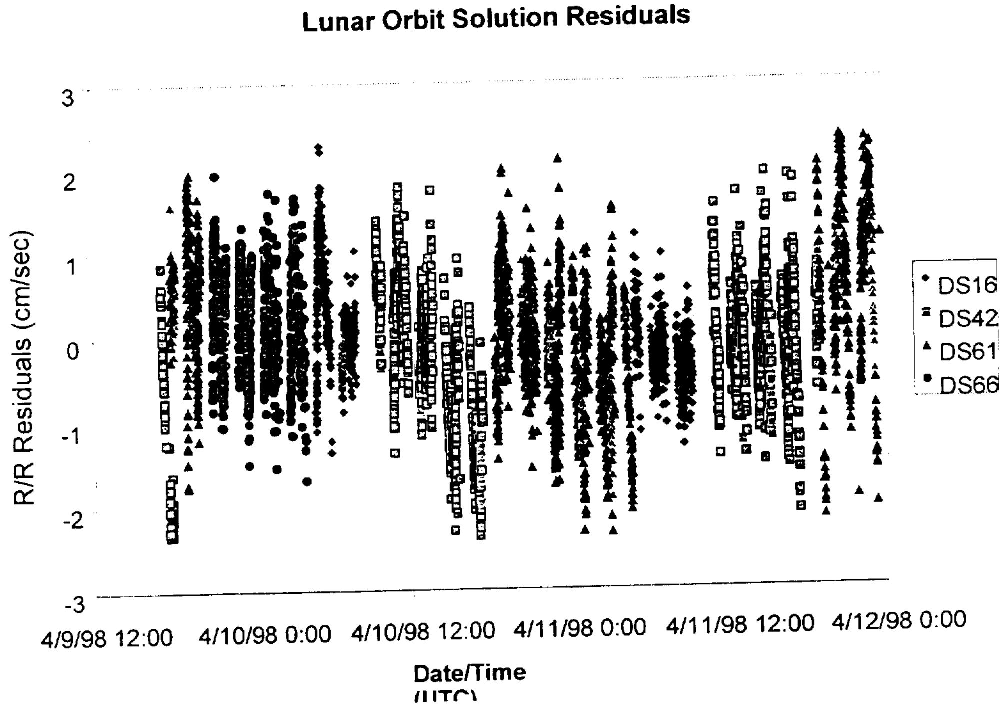


Figure 11: Nominal Mission Solution Residuals

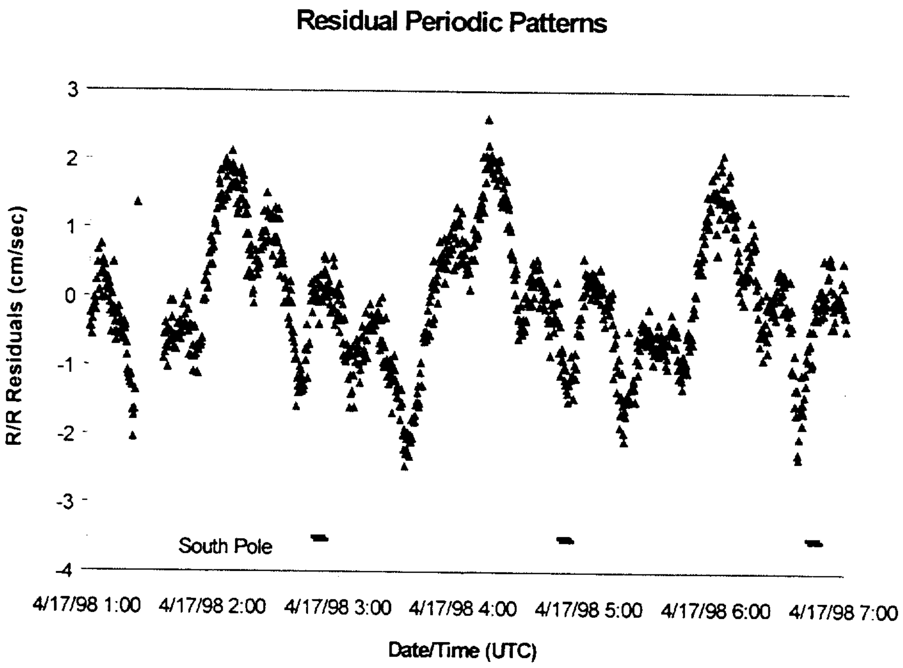


Figure 12: Magnified Nominal Mission Solution Residuals

In order to maintain the nominal lunar mapping orbit, periodic maneuvers were required. These maneuvers and other mission events were planned out weeks in advance based upon long term predicted ephemerides generated from the estimated states. Figure 13 shows the 36-day and 6-day propagation accuracies using the LP75D lunar potential model. The accuracy is determined by propagating from the definitive state for 36 days and comparing to the new definitive state over a 24 hr period. The 36-day predictions were accurate to about 22 km in total position and less than 3 km in radial error. These predictions were updated as mission events and maneuvers approached and the 6-day predictions gave total position errors of less than 500m, with less than 50 m in the radial direction.

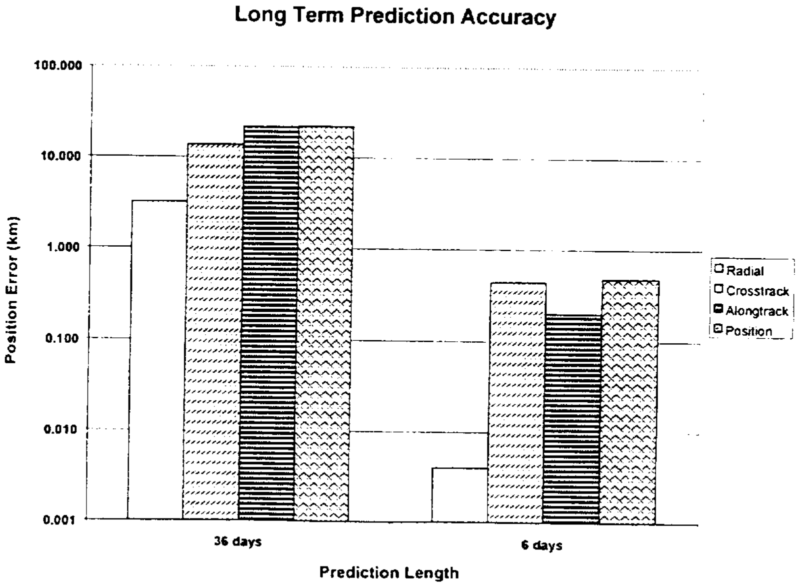


Figure 13: Nominal Mission LP75D Propagation Accuracy

Going from the nominal mission phase to the extended mission phase entailed many changes. The lunar potential model would have to be updated. The prediction accuracies given in Figure 13 are for the 100 km mean orbit and would likely greatly increase at 30 km. The transition 40-km orbit was used to accumulate enough Doppler data to update the potential model and test its prediction capability. The final lunar potential model used for the definitive solutions in the extended mission, LP100J, was developed using all LP data from the nominal 100-km orbit, the transition 40-km orbit, and several weeks worth from the 30-km orbit.

The batch orbit solutions during the extended mission did not change other than the potential model. They still consisted of 55-hr batch solution arcs with 7-hr overlaps. The period of the orbit only changed by about 7 minutes, from 118 minutes to 111 minutes, so the orbit observability did not change much.

The definitive orbit accuracy during the extended mission is shown in Table 5 along with the orbit accuracy from the nominal mission. Note that despite the improved potential model, the orbit accuracy is degraded by a factor of more than four due to the lower altitude. The radial component, the most important for science processing, is still less than 30 meters.

Table 5: Extended Mission Orbit Determination Accuracy

Component	100-km orbit w/ LP75D	30-km orbit w/ LP100J
Radial RMS	13 m	28 m
Crosstrack RMS	155 m	673 m
Alongtrack RMS	189 m	976 m
Position RMS	270 m	1249 m
Avg. Doppler Residual	9.3 mm/sec	4.3 cm/sec
Average Inclination	NA	0.029 deg
Average SMA	NA	106 m

The fit to the Doppler data is also more than a factor of four worse than in the nominal mission phase as seen in Figure 14. This however, does not appear to be solely due to the lunar potential. An evaluation of the residual signature shows that the Doppler noise is considerably higher in the extended mission phase. Figure 15 shows a magnification of the residuals over a shorter 6-hr time frame. This increased noise from the DSN tracking stations is due to the proximity of the spacecraft to the lunar surface.

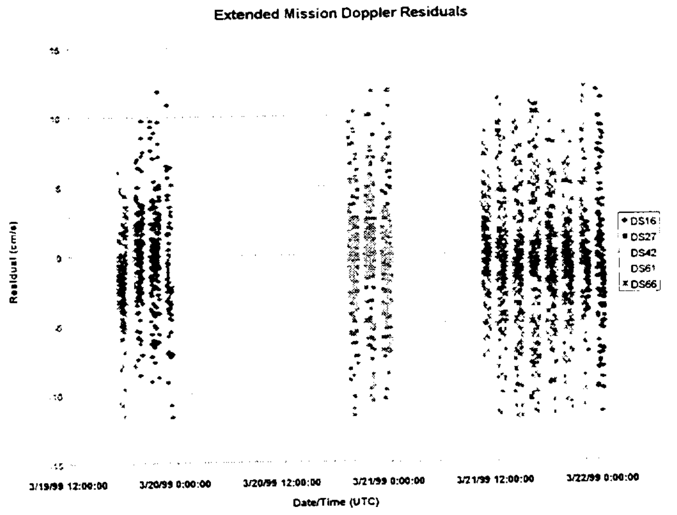


Figure 14: Extended Mission Solution Residuals

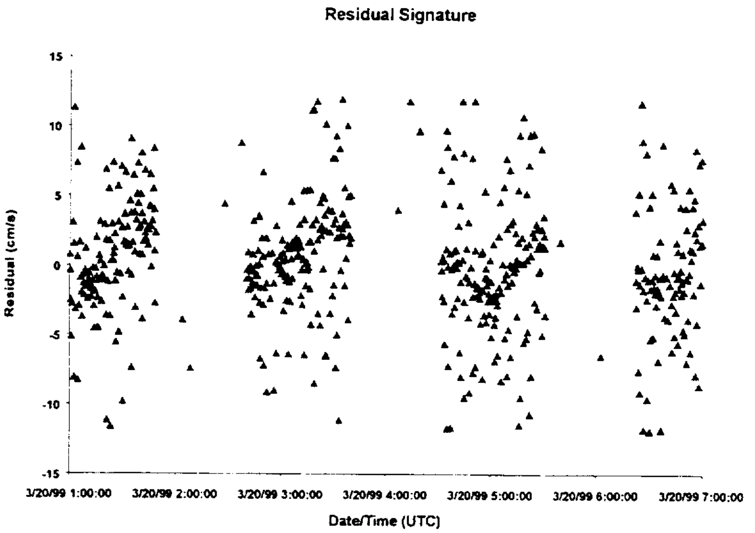


Figure 15: Magnified Extended Mission Solution Residuals

In the extended mission, with the much lower altitudes, it would be important to be able to predict the spacecraft ephemeris with considerable accuracy. At the 30 km mean orbit, maneuvers were planned every 27 days with impact occurring less than a week later. So the entire maneuver schedule for the extended mission depended on the ephemeris prediction accuracy. In the first months of the extended mission, the prediction accuracy was measured during unperturbed mission segments. The prediction accuracies are shown in Figure 16. Even after three weeks of prediction at a 30-km mean orbit, the radial error is less than one km. The alongtrack error however grows to more than 40 km, which translates into 25 seconds in acquisition time.

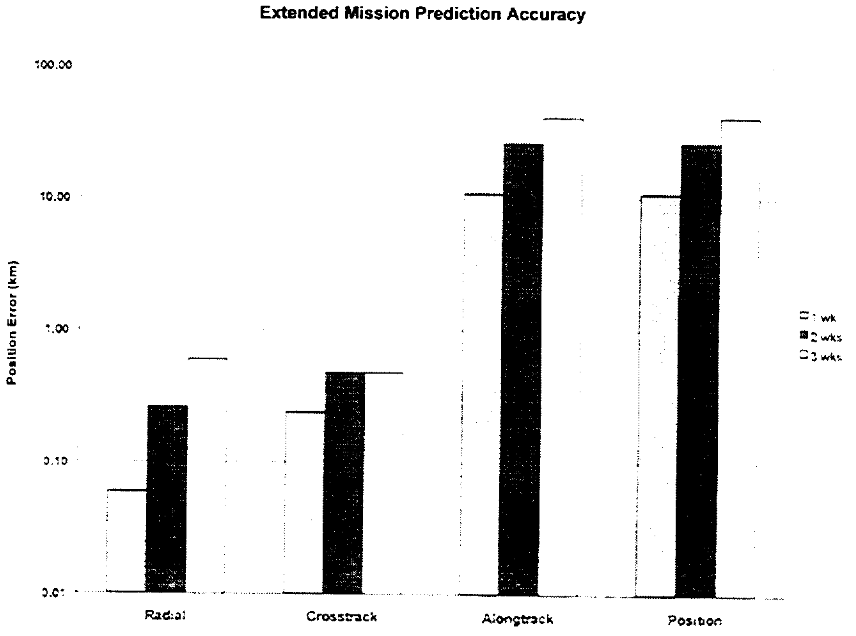
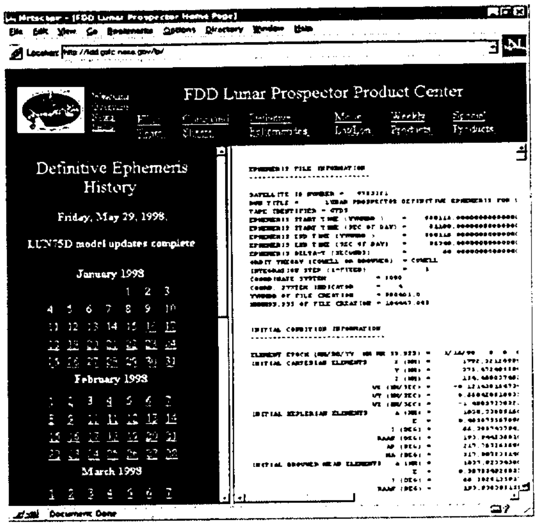


Figure 16: Extended Mission Prediction Accuracy

The GNCC LP Product Center provides the definitive history of the LP mapping orbit, both nominal and extended mission, in two forms: (1) Cartesian ephemeris in the J2000 selenocentric coordinate system and (2) Moon latitude and longitude in a selenographic coordinate system. Each is available for each day of the LP mapping orbit. Figure 17 shows the Definitive Ephemerides page. A calendar of linked dates provides intuitive browsing. To the right of the calendar, the user may browse the file prior to or in lieu of downloading it. The Moon Latitude/Longitude page also uses this format. The GNCC LP Product Center can be found at *http://fdd.gsfc.nasa.gov/lp.*


File Edit View Go Bookmarks Options Directory Window Help
Location: http://fdd.gsfc.nasa.gov/lp/
Document: Done

FDD Lunar Prospector Product Center

Definitive Ephemeris History

Friday, May 29, 1998,

LUN75D model updates complete

January 1998

February 1998

March 1998

EPHEMERIS FILE INFORMATION

INITIAL CONDITION INFORMATION

Figure 17: Definitive Ephemerides Web Page

SUMMARY

The first two months of the extended mission of Lunar Prospector have been a resounding success. Operational results show that selection of orbital parameters need to be considered in the orbit analysis, maneuver planning, and orbit determination. The coupling of the maneuver epoch along with the target orbit eccentricity and argument of periapsis is very important, especially for the low altitudes of the extended mission. The evolution of the eccentricity cannot be eliminated due to the lunar potential, which has been shown to result in oscillations of approximately 0.01. Orbit determination accuracy is very dependent upon the quality and the degree and order of the potential model, with LP O.D. results near 1 km position accuracy. Prediction quality is dependent upon the quality of the definitive and the use of numerical integrators with small step size (e.g <2 minutes). It is therefore recommended to use these findings to place other lunar orbiting spacecraft into quasi-frozen orbits. This will maximize the time between maintenance maneuvers and allows for predictable selenographic altitudes. This practice has been followed for the extended LP mission orbit with very successful results.

REFERENCES

1. Hubbard, G.S. et al., "The Lunar Prospector Discovery Mission: A New Approach to Planetary Science", IAF-97-Q.4.01, Oct. 1997

2. Lozier, Galal, Folta, and Beckman, "Lunar Prospector Mission Design and Trajectory Support", AAS 98-323, GSFC International Flight Mechanics Conference, May 1998

3. Folta, Galal, & David Lozier "Lunar Prospector Frozen Orbit Mission Design" AIAA-98-4288, Astrodynamics Specialist Conference, Boston, MA. 1998

4. Mission Analysis and Design Tool (Swingby) Mathematical Principles (rev 1), 553-FDD-92/058R1UDO, September 1995

5. Konopliv, A.S., "Lunar Prospector Gravity Results – LP100J", as yet unpublished

6. Beckman, M. and Concha, M., "Lunar Prospector Orbit Determination Results", AIAA-98-4561, Astrodynamics Specialist Conference, Boston, MA. 1998

Janus Trajectory Design

Gregory C. Marr
Flight Dynamics Analysis Branch, Code 572, NASA GSFC

ABSTRACT

The proposed Janus Discovery Class mission will pass over opposite illuminated hemispheres of the planet Mercury on two successive flybys and will pass over a crater near Mercury's South geographic pole on the third and final flyby. Three probes will be released by the main spacecraft prior to the first Mercury flyby to pass over Mercury's geographic poles and over the anti-Sunward side. The science team wanted to complete the first Mercury flyby within approximately 110 days of launch and the second Mercury flyby within approximately 365 days of launch. A direct trajectory was chosen which met the basic constraints of the science team while meeting the Discovery launch vehicle constraints. The objective of this paper is to serve as an overview of the Janus trajectory design. The proposal submitted in June 1998 provides a comprehensive overview of the Janus mission.

INTRODUCTION

The Goddard Space Flight Center's (GSFC's) Planetary Magnetospheres Branch proposed a Discovery class mission to Mercury, Janus. The main spacecraft will make three hyperbolic flybys of Mercury. During the first two main spacecraft flybys periapsis will be at approximately 100 km altitude at a low Mercury latitude close to the sub-Solar point. The sub-Solar and sub-periapsis points for the first two main spacecraft flybys will be separated by approximately 180 degrees in Mercury longitude, and the second flyby will occur roughly one year after launch. Different options were explored. A direct trajectory to Mercury followed by a maneuver after first Mercury flyby to increase the Heliocentric orbit period to approximately 264 days, 3 times Mercury's orbital period and 4.5 times Mercury's rotational period, met the science team's objectives within the mass constraints of the available launch vehicles. This direct trajectory was pursued as the baseline trajectory at the recommendation of the science team. The science team preferred the lower elapsed time between launch and first flyby and the less complex operational scenario offered by this option. Prior to the first Mercury flyby two probes or Remote Experiment Packages (REPs) were to be released from the main spacecraft to pass over the North and South geographic poles with periapsis altitudes of approximately 100 km. Later a third REP which would pass anti-Sunward of Mercury at an altitude of approximately 9756 km during the first flyby was added, and a third Mercury flyby by the main spacecraft over a large crater near Mercury's South pole was added (Crater X per reference 2). The main spacecraft will be maneuvered prior to the first Mercury flyby and each REP would be released sequentially with a small fixed delta v by a spring device to decrease the complexity and mass of the REPs.

SCIENCE OBJECTIVES

Per the proposal, Janus was proposed as a fast, low cost, low risk four platform mission to Mercury. The REPs along with the main spacecraft would provide the first multipoint measurements in a planetary magnetosphere. The main spacecraft would provide 1 x 1 km image resolution of the entire planetary surface. The main spacecraft instrument package consists of nine different instruments, and the REP instrument packages consist of three different instruments. The main spacecraft instrument packages are a visible/infrared imager (VIRSI), ultraviolet spectrometer (Alice), X-ray spectrometer (XRS), neutron spectrometer (PHINC), magnetometer (MAG), low energy plasma analyzer (PEPE), energetic particle detector (IPS), search coil (SCX), and electric field instrument (EFI). The REP instrument packages are PEPE, MAG, and SCX.

Mariner 10 was launched November 3, 1973. After a Venus swingby Mariner 10 made three flybys of Mercury on March 29, 1974, at an altitude of about 704 km, on September 21, 1974, at an altitude of about

47,000 km, and on March 16, 1975 at an altitude of about 327 km. The Mariner-10 flybys were separated by approximately 176 days, roughly 2 times Mercury's orbital period. Mercury makes roughly three complete rotations every 176 days. Therefore, the sub-Solar point was at roughly the same Mercury longitude for each Mariner-10 flyby, and the same hemisphere of Mercury was illuminated during each Mariner-10 flyby. Janus has a Heliocentric period of approximately 264 days, roughly 3 times Mercury's orbital period. Mercury makes approximately 4.5 rotations every 264 days. The sub-Solar point will have changed by approximately 180 degrees Mercury longitude between Janus flybys allowing opposite illuminated hemispheres of Mercury to be imaged. Janus periapsis geometry was chosen to meet the science objectives while taking advantage of the Heliocentric period increase after the Mercury swingby to decrease the spacecraft delta v requirement and meet the spacecraft mass constraints.

JANUS BASELINE TRAJECTORY DESIGN

The main spacecraft is 3 axis stabilized. The REPs are spin-stabilized. The estimated total dry mass of the main spacecraft and the 3 REPs is 291.7 kg; the REP mass is 18.6 kg each. The launch vehicle is a Delta II 7925H. The main spacecraft has four 5.0 pound force dual mode thrusters which will nominally use hydrazine and nitrogen tetroxide for orbit maneuvers. In addition eight 0.2 pound force monopropellant thrusters using hydrazine are used for attitude control maneuvers.

C3 is twice the combined kinetic and potential energy per unit mass required of the launch vehicle. DLA is Declination of the Launch or departure Asymptote. Delta v is the velocity change. The minimum C3 opportunities for 2000-2003 period with first Mercury flyby roughly 115 days after launch are listed in Table 1. Table 1 includes the spacecraft delta v requirement for two Mercury flybys (because the third Mercury flyby was not added until later in the analysis) and includes a 120 m/sec allocation for launch vehicle error correction, maneuvers for REP release, spacecraft propulsion system error correction, and attitude control. If the additional 44.3 m/sec delta v required for the third flyby for the 2002 case is added, the total delta v increases to 1427.1 m/sec.

By 2004, the C3 requirement increases to 52.9 km^2/sec^2. Launch opportunities occur roughly every 3 Mercury-Earth synodic periods, with Mercury close to aphelion. The nominal launch opportunity for the Janus proposal submitted in June 1998 was determined to be the 2002 launch opportunity.

The main spacecraft will be maneuvered to place the REPs on the proper trajectory, and the REPs will be released with a small fixed delta v by a spring device at flyby 1 minus 40, 30, and 20 days. After the release of the third REP, the main spacecraft will be maneuvered to its nominal trajectory. This timeline includes an ELV error correction maneuver 10 days after launch but does not include correction of errors in the spacecraft propulsion system. Typically, these error correction maneuvers required a delta v only slightly greater than the error itself. More extensive error correction maneuver analysis was done by GSFC personnel and by JPL personnel as part of the orbit determination and propagation error analysis for which JPL was lead. The hyperbolic excess velocity of the main spacecraft was approximately 12.7 km/sec for flyby 1 and 13.9 km/sec for flybys 2 and 3 resulting in an eccentricity of approximately 20 for flyby 1 and 23 for flybys 2 and 3. For flybys 1 and 2 the sub-periapsis point was within 10 degrees Mercury longitude of the sub-Solar point and was within 20 degrees of the Mercury equator. The trajectories chosen met the objectives of the science team and accounted for the trajectory errors predicted by JPL's orbit determination and orbit propagation error analysis. The North Pole REP passes over the North Pole at an altitude of 382 km with periapsis altitude of 100km. The South Pole REP passes over the South Pole at an altitude of 540 km with periapsis altitude of 100km. The main spacecraft passes over Crater X (Reference 2) on the third flyby at an altitude of 350 km with a periapsis altitude of 100 km. The data above is for the October 17, 2002 launch case, minimum C3.

Figures 1, 2, and 3 show the view from Ecliptic North of the Janus trajectory (and the orbits of Earth and Mercury) for three different stages of the mission leading to Mercury flybys 1, 2, and 3 respectively. Figure 4 shows the view from Ecliptic North of the Janus trajectory (and the orbits of Earth and Mercury) from launch through flyby 3. Figure 5 shows the view from the North pole of Mercury of the 3 main spacecraft flybys; because of software limitations, the poles on the planet in the figure are not the Mercury North and South poles.

The main spacecraft maneuvers and REP releases in a proposed mission timeline for a 2002 minimum C3 launch follow. The epochs are in yyyymmdd/mission elapsed days (ddd) format.
20021017/000: Launch, Minimum C3 (43.0 km^2/sec^2)
20021027/010: ELV error correction maneuver (37.6 m/sec), main spacecraft on anti-Sunward trajectory
20021231/075: REP released to pass anti-Sunward
20030101/076: Main spacecraft maneuvered to North Pole trajectory (5.1 m/s delta v)
20030110/085: REP released to pass over North Pole
20030111/086: Main spacecraft maneuvered to South Pole trajectory (3.8 m/s delta v)
20030120/095: REP released to pass over South pole
20030121/096: Main spacecraft maneuvered to nominal trajectory (3.9 m/s delta v)
20030209/115: Main spacecraft flyby 1
20030211/117: Main spacecraft maneuver to increase Heliocentric period to approximately 264 days for second flyby (1264.9 m/s delta v)
20031030/378: Main spacecraft flyby 2
20031101/380: Main spacecraft maneuver for third encounter (44.3 m/s delta v)
20040720/642: Main spacecraft flyby 3, passage over Crater X near South Pole

There is a 14 day launch window, roughly October 8-26, 2002, which allows for a 30% increase in the spacecraft dry mass and the corresponding increase in fuel mass. The flyby 1 Mercury periapsis epoch remained within 12 hours of the minimum C3 case through the launch window. The trajectory variations were acceptable to the science team.

For the backup direct 2003 launch opportunity with approximately 115 days from launch to first Mercury flyby, the launch vehicle C3 and spacecraft delta v requirements increased such that a 291.7 kg dry mass spacecraft could not meet the mission objectives. A direct solution was found which required approximately 11 months from launch to first Mercury flyby. While this trajectory did not meet the science team's desire for a first Mercury flyby roughly 110 days after launch, it was a viable direct trajectory option.

SUMMARY

The June 1998 submission of the Janus Discovery proposal was the first submission for this mission. Unfortunately, Janus was not one of the proposals selected for further study. It is anticipated this proposal could be resubmitted for a future launch date, and some analysis of those potential launch dates has already been done.

ACKNOWLEDGEMENTS

I want to take this opportunity to acknowledge the outstanding job done by the Principal Investigator, Dr. Steve Curtis. Dr. Curtis functioned extremely effectively both as the science team lead and as the overall study lead. I also acknowledge the very valuable assistance of Julie Kangas and Robert Haw of the Jet Propulsion Laboratory (JPL). Finally, I acknowledge the assistance provided by David Folta of GSFC, Pamela Clark of Catholic University, Barbara Giles of GSFC, Daniel Winterhalter of JPL, and Chris Eyerman of Orbital Sciences Corporation. Launch window analysis was performed using the Mission Analysis Environment (MAnE) software (Adasoft Incorporated) and using GSFC's Swingby software. Detailed trajectory analysis was performed using GSFC's Swingby software and using software developed by the author.

This proposal was completed in partnership with the Jet Propulsion Laboratory. GSFC was lead on the trajectory design, and GSFC's work in this area was checked by JPL personnel. Because of JPL's more extensive experience and superior software (especially for analysis of Heliocentric cruise trajectories) JPL

was lead on the orbit determination error analysis. The trajectories chosen met the objectives of the science team and accounted for the trajectory errors predicted by JPL's orbit determination and orbit propagation error analysis.

Per reference 3, Farquhar and Dunham have proposed a phasing loop approach for interplanetary missions with high launch vehicle energy requirements like this mission. The possible advantages of this approach could be explored in the event of a future submission of the Janus proposal.

TABLES AND FIGURES

Table 1. Janus Minimum C3, DLA, and Spacecraft Delta V Requirements

Departure Date (yymmdd)	C3 (km^2/sec^2) (minimum)	DLA (deg)	Spacecraft Delta V for 2 Mercury flybys (m/sec)
20001119	41.1	-14.7	778.5
20011102	40.8	-16.4	1098.8
20021017	43.0	-21.4	1382.8
20031002	47.3	-25.7	1730.9

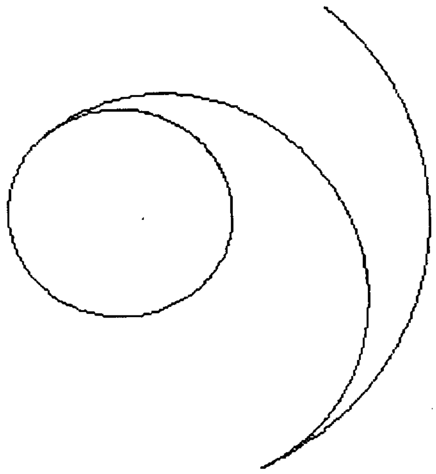


Figure 1: View from Ecliptic North, Launch to Flyby 1 Showing Earth and Mercury Orbits and Janus Trajectory

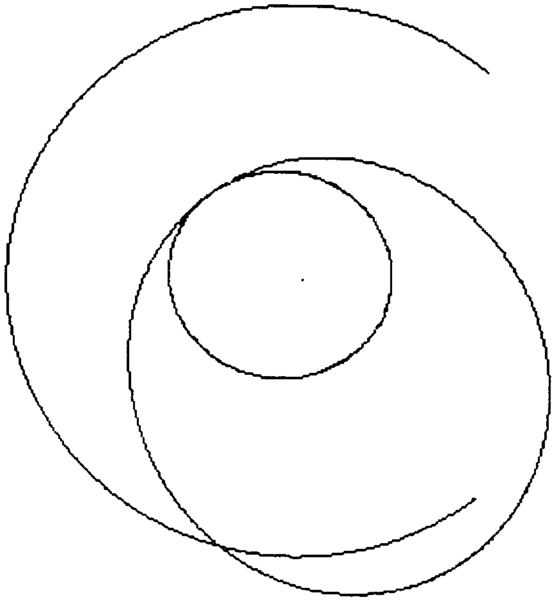


Figure 2: View from Ecliptic North, Flyby 1 to Flyby 2 Showing Earth and Mercury Orbits and Janus Trajectory

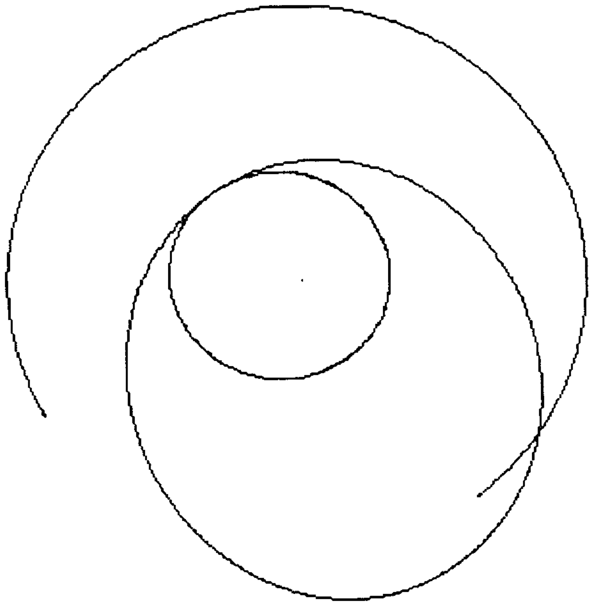


Figure 3: View from Ecliptic North, Flyby 2 to Flyby 3 Showing Earth and Mercury Orbits and Janus Trajectory

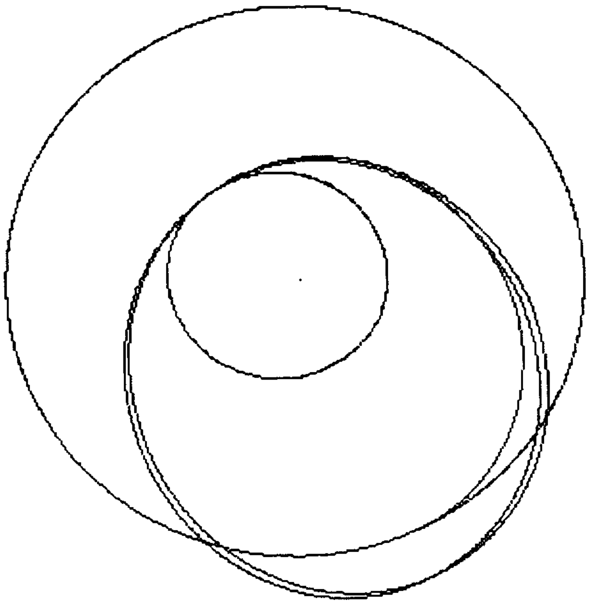


Figure 4: View from Ecliptic North, Launch to Flyby 3 Showing Earth and Mercury Orbits and Janus Trajectory

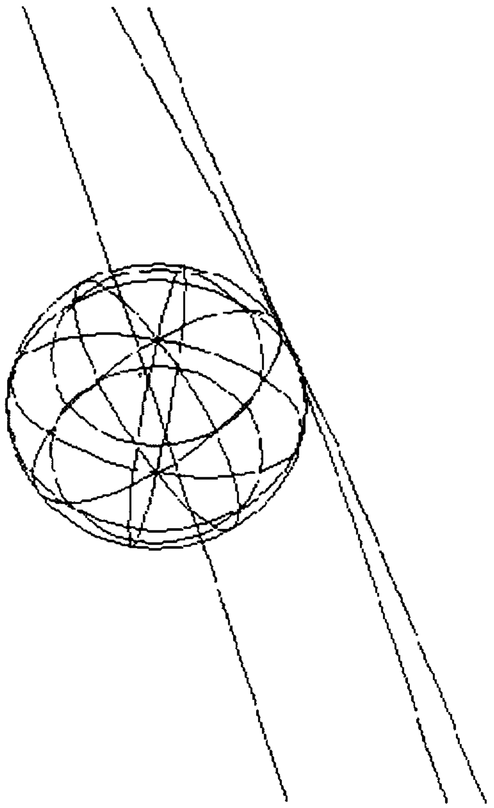


Figure 5: View from Mercury North Geographic Pole of Three Main Spacecraft Mercury Flybys

REFERENCES

1. D. McGiffin and C. Roberts, *Mission Analysis and Design Tool (Swingby) Mathematical Principles*, Revision 1, September 1995

2. M.A. Slade and J.K. Harmon, "Radar Mapping of Mercury's Polar Anomalies", *Nature*, Volume 369, May 19, 1994, page 213

3. R.W. Farquhar and D.W. Dunham, "The Indirect Launch Mode: A New Launch Technique for Planetary Missions", Third IAA Conference on Low-Cost Planetary Missions, April 1998

4. Kaplan, Marshall H., *Modern Spacecraft Dynamics and Control*, 1976

5. Wertz, James R., *Spacecraft Attitude Determination and Control*, 1984

S14-13

J_2 INVARIANT RELATIVE ORBITS FOR SPACECRAFT FORMATIONS*

Hanspeter Schaub and Kyle T. Alfriend
Texas A&M University

ABSTRACT

An analytic method is presented to establish J_2 invariant relative orbits. Working with mean orbit elements, the secular drift of the longitude of the ascending node and the sum of the argument of perigee and mean anomaly are set equal between two neighboring orbits. By having both orbits drift at equal rates on the average, they will not pull apart over time due to the J_2 influence. Two first order conditions are established between the differences in momenta elements (semi-major axis, eccentricity and inclination angle) that guarantee that the drift rates of two neighboring orbits are equal on the average. Differences in the longitude of the ascending node, argument of perigee and initial mean anomaly can be set at will, as long as they are setup in mean element space. For near polar orbits, enforcing both momenta element constraints may result in impractically large relative orbits. It this case it is shown that dropping the equal ascending node rate requirement still avoids considerable relative orbit drift and provides substantial fuel savings.

INTRODUCTION

Earlier studies on the relative motion of spacecraft have used the Clohessy-Wiltshire (CW) equations[1,2] to describe the relative motion. With these equations periodic motion in the relative motion reference frame have been identified. These periodic motions include in-plane, out-of-plane, and combinations of these two motion types. The LISA program[3] has three satellites at 1 AU forming an equilateral triangle in a plane inclined at 60 degrees to the ecliptic. When one includes perturbations, some of these periodic orbits are no longer achievable without control to overcome the deviations. A simple example demonstrates this fact. Consider an out-of-plane relative motion caused by a difference in inclination angles. Due to the J_2 perturbation, the inclination difference will cause a differential nodal precession rate between the two satellites resulting in an oscillatory out-of-plane motion with increasing amplitude. However, the CW equations do not show this motion; they indicate an out-of-plane oscillatory motion with a constant amplitude. To maintain a relative orbit designed with the CW equations, periodic orbit corrections are necessary to cancel deviations caused by the J_2 perturbations. Further, a reference motion and the accompanying state transition matrix might result in an out-of-plane control that changes inclination because the state transition matrix does not indicate the increasing amplitude caused by the inclination difference. For these reasons it is necessary for the reference motion to include at least the J_2 gravitational perturbation effect. The satellites considered are assumed to be equal in size and shape. Therefore the differential drag

*This research was suppurted by the Airforce Office of Scientific Research under Grant F49620-99-1-0075.

effect is of lesser importance in this study than the J_2 effect and is neglected.

The relative orbit geometry between two neighboring satellites is described using the differences in *mean orbit elements*. The advantage of this is that the mean orbit elements change very slowly in the presence of the J_2 perturbation, making it simple to study the long-term behavior of the relative orbit. Brouwer's artificial satellite theory without drag[4] is used to search for J_2 invariant relative orbits. In particular, we seek to match the average drift rates of the two neighboring orbits up to first order, resulting in a closed-path relative motion that is practically invariant to the J_2 perturbations. The advantage of these relative orbits is that they will need very little control to cancel the J_2 effects, and thus require less fuel to maintain.

PROBLEM STATEMENT

At any instant of time, the current inertial position and velocity vectors can be transformed into corresponding instantaneous orbit elements. In the absence of perturbations, these elements are constants. Adding the J_2 perturbation causes the elements to vary according to three types of motion, namely secular growth, short period growth and long period growth. The long period term is the period of the apsidal rotation. Over a short time this looks like a secular growth. The short period growth manifests itself as oscillations of the orbit elements, but doesn't cause the orbits to drift apart. The secular growth is the type of growth that needs to be avoided for relative orbits to be J_2 invariant. This growth is best described through *mean orbit elements.* These are orbit averaged elements which do not show any of the short period oscillations. Mean elements can be obtained analytically or numerically. Highly accurate mean elements that must include atmospheric drag, tesseral harmonic and third body effects probably require numerical averaging. In this paper we use an analytical approach to help determine the accuracy that will be required. By studying the relative motion through the use of mean orbit elements, we are able to ignore the orbit period specific oscillations and address the secular drift directly. It is not possible to set the drift of each orbit to zero. However, instead we choose to set the difference in mean orbit element drifts to zero to avoid *relative secular growth.*

Numerous analytic theories for the motion of an artificial satellite have been developed. The one developed by Shannon Coffey at NRL is the most comprehensive;[5] it has been developed to third order with zonals up to at least J_9. In this study we use the theory developed by Brouwer.[4] We want to look at the motion defined by the mean elements, thus we will use the averaged elements, or in Brouwer's notation, the double-primed elements. This is the Hamiltonian after removal of the short and long period terms. Since $J_n = \mathcal{O}(J_2^2)$ for $n¿2$, the only geopotential effect that is included is J_2.

The orbit geometry is described through the Delaunay orbit elements l (mean anomaly), g (argument of perigee) and h (longitude of the ascending node) with the associated generalized momentas L, G and H defined as

$$L = \sqrt{\mu a} \tag{1a}$$
$$G = \sqrt{\mu a(1-e^2)} = L\eta \tag{1b}$$
$$H = G\cos i \tag{1c}$$

where a is the semi-major axis, e is the eccentricity and i is the inclination angle. The variable η is another convenient variable measuring the eccentricity and is given by

$$\eta = \sqrt{1-e^2} \tag{2}$$

Note that G is the angular momentum of the orbit and H is the corresponding polar component. Unless noted otherwise, any orbit elements used from here on will be assumed to be mean orbit elements. Since the Delaunay variables are canonical variables, given the Hamiltonian $\mathcal{M}$, their rates are found through

$$\dot{l} = \frac{\partial M}{\partial L} \qquad \dot{g} = \frac{\partial M}{\partial G} \qquad \dot{h} = \frac{\partial M}{\partial H} \tag{3a}$$

$$\dot{L} = -\frac{\partial M}{\partial l} \qquad \dot{G} = -\frac{\partial M}{\partial g} \qquad \dot{H} = -\frac{\partial M}{\partial h} \tag{3b}$$

The mean Hamiltonian M can be written as an asymptotic expansion in $\epsilon = -J_2$ as

$$M = M_0 + \epsilon M_1 + \mathcal{O}(\epsilon^2) \tag{4}$$

In this study, we will only focus on the first order terms and ignore higher order terms in ϵ. The first two terms M_0 and M_1 are given by

$$M_0 = -\frac{\mu^2}{2L^2} \tag{5}$$

$$M_1 = -\frac{\mu^4 R_e^2}{4L^6}\left(\frac{L}{G}\right)^3\left(1 - 3\frac{H^2}{G^2}\right) \tag{6}$$

with R_e being Earth's radius at the equator. The following algebra is greatly simplified if we work with dimensionless variables. Therefore distances will be measured in Earth radii R_e and time is normalized by the mean motion of a satellite at one Earth radius (i.e. $\mu = 1$). The dimensionless equivalents of Eqs. (5) and (6) are

$$M_0 = -\frac{1}{2L^2} \tag{7}$$

$$M_1 = -\frac{1}{4L^6}\left(\frac{L}{G}\right)^3\left(1 - 3\frac{H^2}{G^2}\right) \tag{8}$$

The transformation between osculating and mean elements is shown in the appendix. However, note that these transformations become singular for a circular master orbit (i.e. the eccentricity e becomes zero). This causes numerical difficulties in translating mean elements into corresponding osculating elements for near-circular orbits. Future work will study the use of non-singular canonical variables to alleviate this transformation problem.

Since both M_0 and M_1 depend solely on the mean momenta L, G and H (i.e. the angle variables are ignorable), according to Eq. (3b) the mean momenta expressions are constant. Using Eq. (3a), the mean angle rates $\dot{l}$, $\dot{g}$ and $\dot{h}$ are

$$\dot{l} = \frac{1}{L^3} + \epsilon\frac{3}{4L^7}\left(\frac{L}{G}\right)^3\left(1 - 3\frac{H^2}{G^2}\right) \tag{9}$$

$$\dot{g} = \epsilon\frac{3}{4L^7}\left(\frac{L}{G}\right)^4\left(1 - 5\frac{H^2}{G^2}\right) \tag{10}$$

$$\dot{h} = \epsilon\frac{3}{2L^7}\left(\frac{L}{G}\right)^4\left(\frac{H}{G}\right) \tag{11}$$

Since the mean momenta rates $\dot{L}$, $\dot{G}$ and $\dot{H}$ are always zero, we will only be concerned with matching the angle rates between two neighboring orbits in the next section.

CONSTRAINTS FOR J_2 INVARIANT ORBITS

In order to keep two neighboring orbits from drifting apart, the average secular growth needs to be equal. Short period oscillations can be ignored here since these are only "temporary" deviations.

Also, it is assumed that the long period growth (due to the rotation of the orbit plane) is very slow. The following development does not guarantee that the long period growth will necessarily be equal.

Since the mean angle quantities l, g and h do not directly contribute to the secular growth caused by J_2, their values can be chosen at will. However, the mean momenta values L, G and H (and therefore implicitly a, e and i) must be carefully chosen to match the secular drift rates.

To keep two neighboring orbits from drifting apart over time, one requirement is that the mean nodal rates $\dot{h}_i$, defined in Eq. (11), must be equal.

$$\dot{h}_i = \dot{h}_j \qquad \forall i \neq j \tag{12}$$

However, it is not necessary that the average mean anomaly rate $\dot{l}_i$ or argument of perigee rate $\dot{g}_i$ are equal. For example, it is possible for the average mean anomaly difference to increase, while the average argument of perigee difference decreases. Even though the lines of perigee are drifting apart over time, the two spacecraft could still remain close if the difference in mean anomaly drift can compensate for it. Thus, instead of setting the $\dot{l}_i$ and $\dot{g}_i$ equal, we set the sum of the two rates equal

$$\dot{\theta}_i = \dot{l}_i + \dot{g}_i = \dot{\theta}_j \qquad \forall i \neq j \tag{13}$$

where θ called the latitude angle. Combining Eqs. (9) and (10), the latitude rate $\dot{\theta}$ is expressed as

$$\dot{\theta} = \frac{1}{L^3} + \epsilon \frac{3}{4L^7} \left(\frac{L}{G}\right)^3 \left[\left(1 - 3\frac{H^2}{G^2}\right) + \frac{L}{G}\left(1 - 5\frac{H^2}{G^2}\right)\right] \tag{14}$$

Let the reference mean orbit elements be denoted with the subscript "0". The drift rate $\dot{\theta}_i$ of a neighboring orbit can be written as a series expansion about the reference orbit element as

$$\dot{\theta}_i = \dot{\theta}_0 + \frac{\partial \dot{\theta}_0}{\partial L}\delta L + \frac{\partial \dot{\theta}_0}{\partial G}\delta G + \frac{\partial \dot{\theta}_0}{\partial H}\delta H + H.O.T. \tag{15}$$

where we make use of the fact that $\dot{\theta} = \dot{\theta}(L, G, H)$ only. Let the difference in latitude rates be $\delta\dot{\theta}$, then a first order approximation of Eq. (15) is written as

$$\delta\dot{\theta} = \dot{\theta}_i - \dot{\theta}_0 = M_{\theta L}\delta L + M_{\theta G}\delta G + M_{\theta H}\delta H \tag{16}$$

where $M_\theta = M_l + M_g$ and

$$M_{\alpha A} = \left.\frac{\partial \dot{\alpha}}{\partial A}\right|_{L=L_0, G=G_0, H=H_0} \tag{17}$$

Similarly, we can expand the nodal rate $\dot{h}$ to find

$$\delta\dot{h} = M_{hL}\delta L + M_{hG}\delta G + M_{hH}\delta H \tag{18}$$

To enforce equal drift rates $\dot{\theta}_i$ and $\dot{h}_i$ between neighboring orbits, we must set $\delta\dot{\theta}$ and $\delta\dot{h}$ equal to zero in Eqs. (16) and (18), resulting in the following two necessary conditions for relative orbits to be J_2 invariant up to first order.

$$M_{\theta L}\delta L + M_{\theta G}\delta G + M_{\theta H}\delta H = 0 \tag{19}$$

$$M_{hL}\delta L + M_{hG}\delta G + M_{hH}\delta H = 0 \tag{20}$$

Since Eqs. (19) and (20) have three unknown quantities, namely the differences in mean momenta δL, δG and δH, we are only left with one degree of freedom in selecting the relative momenta. After choosing either δL, δG or δH, the remaining two momenta differences are determined through the two conditions shown above. If we choose to enforce that the various $\dot{l}_i$ and $\dot{g}_i$ are equal, then we have a nonlinear system of three variables with three constraints, leaving no degree of freedom in choosing the momenta quantities a, e and i. These conditions are only satisfied for particular orbit solutions which are of little interest for spacecraft formation flying.

We can work in either the (L, G, H) or (a, e, i) space. The first is easier, the latter is more intuitive. Since the mean anomaly, argument of perigee and right ascension are ignorable coordinates in the mean element space, they have no effect on the secular rates between two objects. In the osculating space they are not ignorable and consequently have an effect on the secular relative motion. The angles differences δl, δg and δh can therefore be chosen at will. Thus, operating in mean element space has reduced the scope of the problem. This leaves us with a total of four degrees of freedom to design a J_2 invariant relative orbit.

Using Eqs. (9) through (11), the required partial derivatives are found to be:

$$M_{lL} = \frac{\partial \dot{l}}{\partial L} = -\frac{3}{L^4} + \epsilon\frac{3}{L^8}\left(\frac{L}{G}\right)^3\left(-1 + 3\frac{H^2}{G^2}\right) \tag{21a}$$

$$M_{lG} = \frac{\partial \dot{l}}{\partial G} = \epsilon\frac{9}{4L^8}\left(\frac{L}{G}\right)^4\left(-1 + 5\frac{H^2}{G^2}\right) \tag{21b}$$

$$M_{lH} = \frac{\partial \dot{l}}{\partial H} = -\epsilon\frac{9}{2L^8}\left(\frac{L}{G}\right)^4\left(\frac{H}{G}\right) \tag{21c}$$

$$M_{gL} = \frac{\partial \dot{g}}{\partial L} = \epsilon\frac{9}{4L^8}\left(\frac{L}{G}\right)^4\left(-1 + 5\frac{H^2}{G^2}\right) \tag{22a}$$

$$M_{gG} = \frac{\partial \dot{g}}{\partial G} = \epsilon\frac{3}{2L^8}\left(\frac{L}{G}\right)^5\left(-2 + 15\frac{H^2}{G^2}\right) \tag{22b}$$

$$M_{gH} = \frac{\partial \dot{g}}{\partial H} = -\epsilon\frac{15}{2L^8}\left(\frac{L}{G}\right)^5\left(\frac{H}{G}\right) \tag{22c}$$

$$M_{hL} = \frac{\partial \dot{h}}{\partial L} = -\epsilon\frac{9}{2L^8}\left(\frac{L}{G}\right)^4\left(\frac{H}{G}\right) \tag{23a}$$

$$M_{hG} = \frac{\partial \dot{h}}{\partial G} = -\epsilon\frac{15}{2L^8}\left(\frac{L}{G}\right)^5\left(\frac{H}{G}\right) \tag{23b}$$

$$M_{hH} = \frac{\partial \dot{h}}{\partial H} = \epsilon\frac{3}{2L^8}\left(\frac{H}{G}\right) \tag{23c}$$

Using Eqs. (21) and (22), we are able to rewrite Eq. (19) which enforces equal latitude rates.

$$\begin{aligned}
-\frac{3}{{L_0}^4}\delta L + \epsilon\frac{3}{4L_0^8}\left(\frac{L_0}{G_0}\right)^3\left[-\left(4 + 3\frac{L_0}{G_0}\right) + 3\left(4 + 5\frac{L_0}{G_0}\right)\left(\frac{H_0}{G_0}\right)^2\right]\delta L \\
+ \epsilon\frac{3}{4L_0^8}\left(\frac{L_0}{G_0}\right)^4\left[-\left(3 + 4\frac{L_0}{G_0}\right) + 15\left(1 + 2\frac{L_0}{G_0}\right)\frac{H_0^2}{G_0^2}\right]\delta G \\
- \epsilon\frac{3}{2L_0^8}\left(\frac{L_0}{G_0}\right)^4\left(\frac{H_0}{G_0}\right)\left(3 + 5\frac{L_0}{G_0}\right)\delta H = 0
\end{aligned} \tag{24}$$

Note that only the term δL appears without being multiplied by the small parameter ϵ. Thus δL must be itself of $\mathcal{O}(\epsilon)$ and the term involving $\epsilon\delta L$ can be dropped as a higher order term. The first necessary condition is then written as

$$-\frac{3}{L_0{}^4}\delta L + \epsilon\frac{3}{4L_0^8}\left(\frac{L_0}{G_0}\right)^4\left[-\left(3+4\frac{L_0}{G_0}\right)+15\left(1+2\frac{L_0}{G_0}\right)\frac{H_0^2}{G_0^2}\right]\delta G$$
$$-\epsilon\frac{3}{2L_0^8}\left(\frac{L_0}{G_0}\right)^4\left(\frac{H_0}{G_0}\right)\left(3+5\frac{L_0}{G_0}\right)\delta H = 0 \quad (25)$$

Using the partial derivatives defined in Eq. (23), we are able to rewrite the second condition for J_2 invariant orbits, given in Eq. (20), as

$$\epsilon\frac{3}{2L_0^8}\left(\frac{L_0}{G_0}\right)^4\left[-3\frac{H_0}{G_0}\delta L - 5\frac{H_0}{G_0}\frac{L_0}{G_0}\delta G + \frac{L_0}{G_0}\delta H\right] = 0 \quad (26)$$

Since $\delta L = \mathcal{O}(\epsilon)$ the δL term is dropped, resulting in the greatly simplified condition

$$\delta H = 5\frac{H_0}{G_0}\delta G = 5\cos i_0 \delta G \quad (27)$$

which enforces equal nodal rates $\dot{h}$. Using the δH defined in Eq. (27), we are able to simplify the condition in Eq. (25) to

$$\delta L = \underbrace{-\frac{\epsilon}{4L_0^4}\left(\frac{L_0}{G_0}\right)^5\left(4+3\frac{G_0}{L_0}\right)\left(1+5\left(\frac{H_0}{G_0}\right)^2\right)}_{D}\delta G \quad (28)$$

Combined, Eqs. (27) and (28) provide the two necessary conditions on the mean momenta differences between two neighboring orbits to yield a J_2 invariant relative orbit.

For more physical insight into these constraints, it is convenient to map them into differences in the semi-major axis a, eccentricity measure η and the inclination angle i. The reason for choosing to deal with variations in η and not the eccentricity measure e itself will become clear shortly. Recalling that $L = \sqrt{a}$ (L is a non-dimensional variable), the variations in L and a are related through

$$\delta L = \frac{1}{2L}\delta a = \frac{\delta a}{2\sqrt{a}} \quad (29)$$

Using $G = L\eta$, the variation of G is rigorously

$$\delta G = \delta L\eta + L\delta\eta \quad (30)$$

However, since $\delta L = \mathcal{O}(\epsilon)$, we may drop this term to approximate δG as

$$\delta G \approx L\delta\eta \quad (31)$$

Since $H = G\cos i$, the variation of the polar angular momentum component is

$$\delta H = \delta G\cos i - G\sin i\delta i \quad (32)$$

Substituting Eqs. (29) and (31) into Eq. (28), this constraint enforcing equal latitude rates between two orbits is rewritten as

$$\delta a = 2Da\delta\eta \quad (33)$$

Note that this a is the non-dimensional semi-major axis and must be multiplied by the Earth radius R_e to obtain proper physical units. Substituting Eqs. (31) and (32) into Eq. (27), the constraint enforcing equal nodal rates $\dot{h}$ is rewritten as

$$\delta\eta = -\frac{\eta}{4}\tan i\delta i \quad (34)$$

Combined, Eqs. (33) and (34) form the two necessary momenta constraints expressed in terms of a difference in semi-major axis, eccentricity and inclination angle.

To write the constraint in Eq. (34) in terms of the eccentricity e directly, we must take the first variation of $\eta = \sqrt{1-e^2}$.

$$\delta e = -\frac{\eta}{e}\delta\eta \tag{35}$$

Substituting this into Eq. (34) we find

$$\delta e = \frac{(1-e^2)\tan i}{4e}\delta i \tag{36}$$

Clearly numerical difficulties arise with this constraint expression whenever $e \to 0$ and the reference orbit becomes circular. According to Eq. (36), it would appear that the change in eccentricity required for a given δi would grow infinitely large as e becomes zero. However, Eq. (34) shows that this is not necessary. Using η as the eccentricity measure, we find that the change in eccentricity reaches a finite limit for a circular orbit. The reason for this discrepancy is that it is not the constraint condition that causes the singularity, but the transformation between variations in e and η in Eq. (35). To avoid numerical difficulties with circular reference orbits, it is therefore convenient to describe necessary changes in eccentricity through $\delta\eta$ and then use the nonlinear mapping $e = \sqrt{1-\eta^2}$ to compute the adjusted eccentricity.

Note that Eq. (34) shows a fundamental limitation of these mean momenta constraints. For near-polar orbits, where the inclination angle is close to 90 degrees, the $\tan i$ term grows very large. Even a small change in inclination angle δi, typically done to achieve out-of-plane relative motion, would result in a relatively large change in eccentricity. The result is that the resulting J_2 invariant relative orbits grow very large for near-polar orbits, making these orbits of little practical use for close formation flying applications. However, note that the three mean angle variables can still be picked at random without causing any orbit drift, even for the polar case. Further, note that if a change in eccentricity is prescribed for a near-polar orbit, the associated required change in inclination angle would be very small. Thus enforcing equal drift rate conditions for near-polar orbit only encounter practical difficulties if a particular change in orbit inclination angle is demanded. As numerical simulations will show, setting up this worst case problem in mean element space and then transforming to corresponding inertial position and velocity vectors will typically still exhibit less secular drift than if the problem is simply setup using osculating elements. Further, while it won't be possible to perfectly compensate for the ascending node drift difference due to δi, it is still possible to equalize the latitude rate drifts $\dot{\theta}_i$ using Eq. (33).

NUMERICAL SIMULATIONS

Two numerical studies are presented illustrating J_2 invariant relative orbits for both non-polar and near-polar master orbits. The orbit elements for the master orbit are the same for each simulation except for the inclination angle i as shown in Table I. The orbit has an altitude of 775 km. Since each spacecraft is assumed to be of equal type, differential drag effects are ignored here. The purpose of these simulations is to illustrate how well the first order conditions in Eq. (33) and Eq. (34) render the resulting relative orbit J_2 invariant. Further, the power of setting up relative orbits in terms of mean orbit elements vs osculating element space is shown.

The relative orbit is constructed in these simulations by choosing particular differences in mean orbit elements, and then translating the adjusted mean orbit elements of the second satellite into corresponding osculating orbit elements. The numerical simulation then uses the corresponding initial position and velocity vector and solves the system using the nonlinear equations of motion including the J_i terms up to fifth order. However, in all cases tested the inclusion of the J_3 through J_5 terms had an minimal effect on the answer.

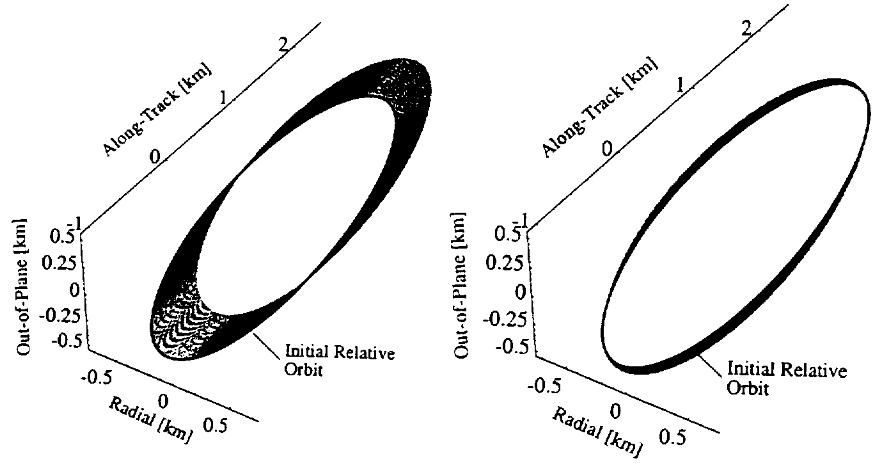
TABLE I
MASTER SATELLITE ORBIT ELEMENTS

Mean Orbit Elements	Value	Units
a	7153	km
e	0.05	
i	48 or 88	deg
h	0.0	deg
g	30.0	deg
l	0.0	deg

Other techniques could be used as well to setup the initial relative orbit. For example, it is possible to establish a desired relative orbit using the natural solutions of the linear CW equations. Since this is the desired *mean* behavior, these six position and velocity coordinates must then be first translated into six corresponding mean orbit elements. Thus we are able to make use of the analytic first order transformation shown in the appendix to obtain the corresponding initial osculating elements (i.e. actual position and velocity vector).

Non-Polar Master Orbit

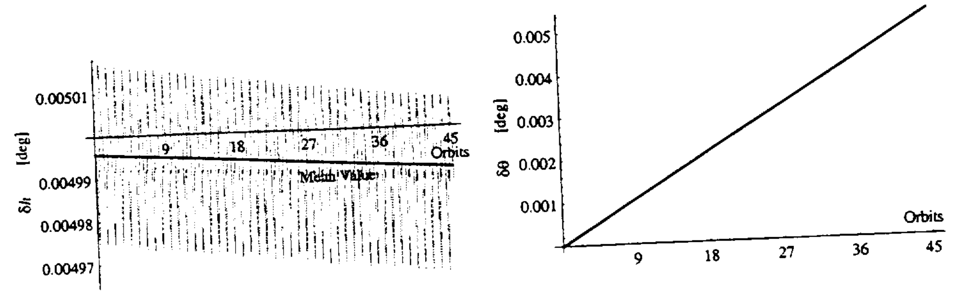
The first simulation illustrates how well the matching conditions work for non-polar orbits. Here the inclination angle is set to 48 degrees. The relative orbit is described by choosing the following mean orbit element differences. To achieve some out-of-plane motion, a ascending node difference of δh = 0.005 degrees is prescribed. The line of perigee and initial mean anomaly differences are set equal and opposite in sign as δg = 0.01 degrees and δl = -0.01 degrees. Of the three momenta elements, we chose to prescribe a change in eccentricity δe = 0.0001 to exaggerate the in-plane, relative orbit. Using Eqs. (33) and (34), the corresponding changes in a and i must be δa = -0.351765 meters and δi = 0.001035 degrees. Note that both the required δa and δi to compensate for this δe are rather small.



(a) Initial Relative Orbit Setup in Osculating Elements

(b) Initial Relative Orbit Setup in Mean Elements

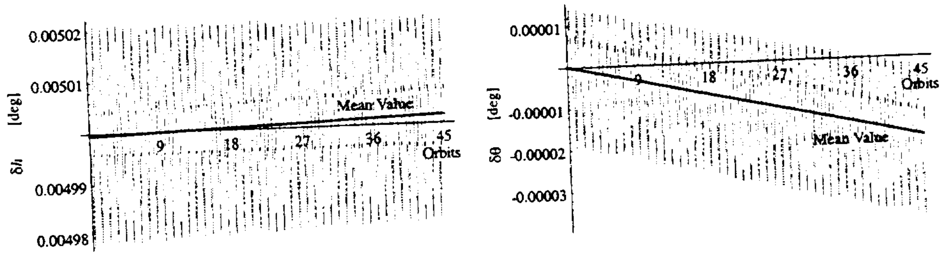
Figure 1 Relative Orbit Drift for a Non-Polar Master Orbit



(a) Difference in Ascending Node h

(b) Difference in Latitude Angle θ

Figure 2 Mean and Osculating Orbit Element Differences for Osculating Element Space Setup



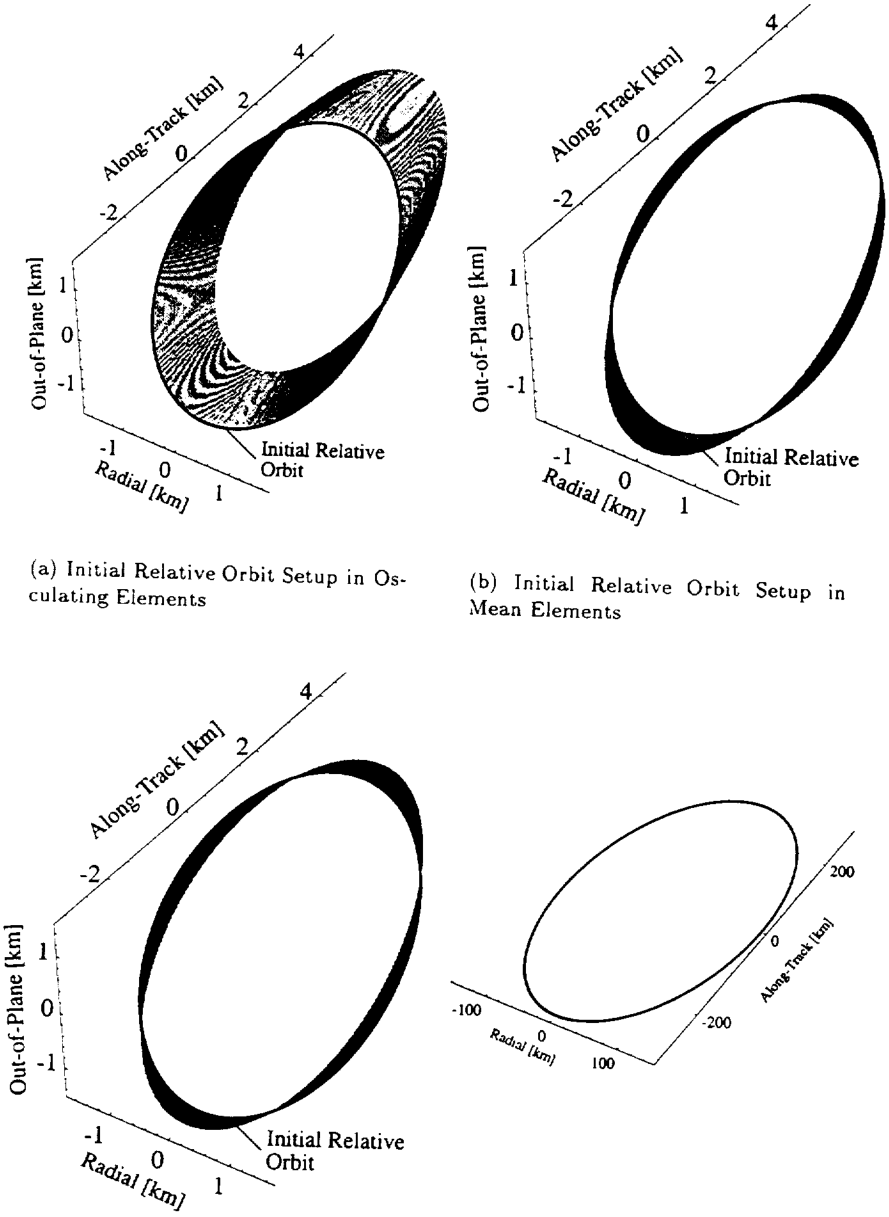
(a) Difference in Ascending Node h

(b) Difference in Latitude Angle θ

Figure 3 Mean and Osculating Orbit Element Differences for Mean Element Space Setup

The resulting relative orbits, as seen in the LVLH frame, are shown in Figure 1. The plots always show the data of 45 orbits, which correspond to roughly 3 days of simulation time. The LVLH frame is chosen such that the $\hat{x}$ direction is along the instantaneous master satellite position vector. The out-of-plane component $\hat{z}$ is found by computing the cross product of $\hat{x}$ with the normalized velocity vector. The along track $\hat{y}$ direction is then found by taking the cross product of $\hat{z}$ and $\hat{x}$. The initial relative orbit is always shown as a solid black line, while the path of the remaining 45 orbits is shown as a gray line. Both simulations use the same initial orbit element differences. In Figure 1(a) the initial orbit element differences, which determine the initial shape of the relative orbit, are chosen in osculating element space. Substantial relative orbit drift is apparent due to the perturbative influence of J_2. Figure 1(b) illustrates the drastic improvements that may occur if the initial orbit geometry is setup in mean element space. Since the matching conditions in Eq. (33) and (34) are only up to first order, the relative orbit will not necessarily be perfectly J_2 invariant. While some periodic thrusting is still necessary, the frequency of these orbit corrections can be greatly reduced.

The differences of δh and $\delta\theta$ between the master and secondary orbits are shown in Figure 2 for the case where the initial setup is performed in the osculating element space. The mean orbit elements are shown as a solid black line, while the osculating elements are shown as a gray line. The corresponding orbit element differences are shown in Figure 3 for the case where the setup is performed in mean element space. While for an inclination angle of 48 degrees both orbits experience a substantial nodal rate $\dot{h}$, the *difference* in ascending node rates is rather small. Setting up the relative geometry in mean element space does reduce the relative nodal drift, but not substantially. A rough calculation of the Δv required per year to compensate for this drift shows 0.0725 m/s required for the osculating element setup, and only 0.0181 m/s required for the mean element setup. Both are relatively small numbers. At this inclination, the $\dot{\theta}$ drift is the dominant factor pulling



(a) Initial Relative Orbit Setup in Osculating Elements

(b) Initial Relative Orbit Setup in Mean Elements

(c) Initial Relative Orbit Setup in Mean Elements Using $\delta a = 2Da\delta\eta$

(d) Initial Relative Orbit Setup in Mean Elements Using Both Matching Conditions

Figure 4 Relative Orbit Drift for a Near-Polar Master Orbit

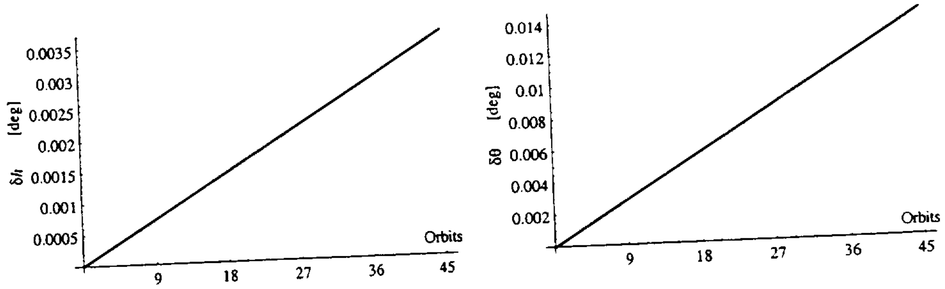
the two orbits apart. Comparing Figures 2(b) and 3(b) the benefit of using mean elements is clear. Using the osculating setup, the Δv required per year is roughly 40.15 m/s. Using the mean orbit elements to setup the geometry reduces this to 0.145 m/s. Using the momenta element matching condition and working in mean orbit element space, we are able substantially reduce the J_2 induced drift and the corresponding Δv's required to reset the relative orbits.

Near-Polar Master Orbit

The second simulation illustrates some issues that arise when trying to generate J_2 invariant relative orbits for near-polar master orbits and demanding a specific inclination angle difference for out-of-plane motion. The inclination angle is set to 88 degrees for this purpose. The relative orbit is described by choosing the mean orbit element differences δh = 0.0 degrees (all out-of-plane motion produced through δi), δg = 0.1 degrees and δl = -0.1 degrees. Assume the relative orbit geometry requires a δi of 0.01 degrees to achieve roughly 1 km of out-of-plane motion. However, we are no longer able to use both matching conditions in Eqs. (33) and (34) since the $\tan i$ term will result in unpractically large changes in eccentricity. Therefore we abandon the hope to be able to compensate for the δh drifts. For near polar orbits, even though the various $\dot{h}$ rates are relatively small, the *differences* of these rates between neighboring orbits with different inclination angles are large. However, we are still able to use Eq. (33) to match latitude drift rates. Therefore we are left with one unused degree of freedom and choose a δe of 0.0001 to exaggerate the in-plane relative orbit.

As the illustrations in Figure 4 show, the J_2 induced drift can still be reduced by simply setting up the relative geometry in mean element space. Figure 4(a) illustrates the motion resulting from setting up the desired orbit element differences in osculating orbit space. The relative orbits pull apart substantially in three days. Figure 4(b) shows the reduced amount of drift that occurs if the same orbit element differences are setup in mean element space. Note that Eq. (33) has not been utilized here to compensate for the latitude difference drift. The relative orbit is thus seen to drift in the negative along track direction. In Figure 4(c) the semi-major axis a is adjusted using Eq. (33) to attempt to equalize the latitude rates $\dot{\theta}$. The required δa is -0.24157 meters. While there is still some drift in the relative orbits due to the different $\dot{h}$ rates, the orbits no longer pull part due to different latitude rates. Figure 4(d) shows how the relative orbit may become excessively large if we attempt to cancel all relative orbit drift for near-polar orbits. To achieve a desired δi of 0.01 degrees, the other two momenta elements differences must be δe = 0.020648 degrees and δa = -27.2122 meters. While the resulting near-polar relative orbit has essentially no drift as seen in this scale, the relative orbit radius grows from a few kilometers to over 100 kilometers. Note that the desired ±1 km out-of-plane motion isn't even visible on the scale shown.

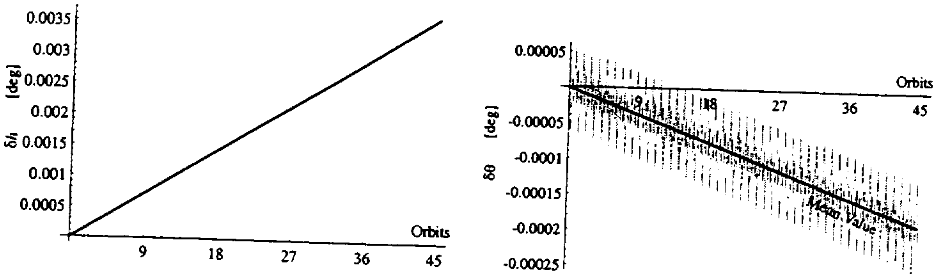
The differences in ascending node and latitude angles for the cases where the relative orbit geometry is setup in the osculating space and where it is setup in the mean element space with semi-major axis adjustment are shown in Figures 5 and 6. As predicted, the ascending node drift δh is the same for both cases since we are no longer trying to compensate for this. Over a year, the Δv required to compensate for this drift is roughly 56.8 m/s. However, where the osculating element setup results in a substantial latitude drift $\delta\theta$, setting up the orbits in mean element space *and* compensating through a corresponding δa results in a substantially reduced latitude drift. The Δv requirement



(a) Difference in Ascending Node h

(b) Difference in Latitude Angle θ

Figure 5 Mean and Osculating Orbit Element Differences for Osculating Element Space Setup for a Near-Polar Master Orbit



(a) Difference in Ascending Node h

(b) Difference in Latitude Angle θ

Figure 6 Mean and Osculating Orbit Element Differences for Mean Element Space Setup for a Near-Polar Master Orbit

to compensate for the $\delta\theta$ is approximately 112 m/s for the osculating setup. This Δv drops to 14.1 m/s if the orbit elements are setup in mean element space. The Δv requirement per year is then further reduced to approximately 1.45 m/s if the δa adjustment is made to equalize the averaged latitude rates.

While this method is not able to compensate for the δh drift encountered with near-polar orbits, it is possible to establish an approximate solution that greatly reduces the J_2 induced relative orbit drift. Note that prescribing differences to h, g and l is always possible, even for polar master orbits. Problems may arise when trying to match a, e and i for a prescribed difference in one of the quantities.

CONCLUSION

A method is presented to establish J_2 invariant relative orbits for spacecraft formation flying applications. The desired relative orbit geometry is designed using differences in mean orbit elements. Two constraints on the three momenta element differences δa, δe and δi are derived. These leave a total of four degrees of freedom to design the relative orbit. As the inclination angle i approaches a polar orbit, the corrections required in eccentricity and semi-major axis to compensate for the J_2 effect become too large to be of practical value. Working with near-polar orbits, setting up the relative orbit geometry in mean elements and canceling the latitude rate difference up to first order still provides a potentially substantial drift and associated fuel savings. A particular limitation of the presented method is that the mapping between mean and osculating elements goes singular for circular orbits. The momenta element differences constraints still hold for circular orbits, but the mapping from mean to osculating elements has mathematical problems whenever e approaches zero. Future work will attempt to rectify this by performing the transformation using non-singular, canonical orbit elements.

REFERENCES

1. T. E. Carter, "State Transition Matrix for Terminal Rendezvous Studies: Brief Survey and New Example," *Journal of Guidance, Navigation and Control*, 1998, pp. 148–155.

2. R. Sedwick, D. Miller, and E. Kong, "Mitigation of Differential Perturbations in Clusters of Formation Flying Satellites," *AAS/AIAA Space Flight Mechanics Meeting*, February 1999. Paper No. AAS 99-124.

3. "LISA:Laser Interferometer Space Antenna," tech. rep., European Space Agency, February 1996. MPQ 208.

4. D. Brouwer, "Solution of the Problem of Artificial Satellite Theory Without Drag," *The Astronautical Journal*, Vol. 64, No. 1274, 1959, pp. 378–397.

5. S. L. Coffey and A. Deprit, "Third Order Solution for Artificial Satellites," *AIAA Journal of Guidance, Control and Dynamics*, July-Aug. 1982.

Appendix

TRANSFORMATION BETWEEN OSCULATING AND MEAN ELEMENTS

For completeness, this appendix illustrates the transformation between osculating and mean orbit elements. This is accomplished through two transformations. The first transformation maps osculating elements to intermediate or long period elements, while the second transformation maps long period elements to mean elements. Using Brouwer's notation, the long period elements are denoted with a single prime, while the mean elements are denoted with a double prime. The generating function W_1^{sp}, which establishes the osculating to long period elements transformation, is given in terms of non-dimensional Delaunay variables as

$$W_1^{sp} = \frac{1}{4G^3}\left(\left(-1+3\frac{H^2}{G^2}\right)(f-l+e\sin f) + \frac{3}{2}\left(1-\frac{H^2}{G^2}\right)\left(\sin(2f+2g)+e\sin(f+2g)+\frac{e}{3}\sin(3f+2g)\right)\right) \tag{37}$$

The notation W_1^{sp} says that this is a first order transformation which removes the short period (sp) component. In Reference 5, Coffey accomplishes this with two transformations. The generating function W_1^{lp}, which establishes the long period to mean elements transformation, is given by

$$W_1^{lp} = -\frac{1}{32G^3}\left(1-\frac{G^2}{L^2}\right)\left(1-5\frac{H^2}{G^2}\right)^{-1}\left(1-16\frac{H^2}{G^2}+15\frac{H^4}{G^4}\right)\sin 2g \tag{38}$$

The transformation between long period and osculating elements is achieved through

$$L' = L - \epsilon(L, W_1^{sp}) = L + \epsilon\frac{\partial W_1^{sp}}{\partial l} \tag{39}$$

$$l' = l - \epsilon(l, W_1^{sp}) = l - \epsilon\frac{\partial W_1^{sp}}{\partial L} \tag{40}$$

with analogous transformations for the other momenta and angle orbit elements. The expression (L, W_1^{sp}) is the Poisson bracket of L and W_1^{sp}. The inverse for this transformation is achieved trivially by switching the primed and unprimed letters and reversing the sign of the ϵ term.

$$L = L' + \epsilon(L', W_1^{sp}) = L - \epsilon\frac{\partial W_1^{sp}}{\partial l'} \tag{41}$$

$$l = l' + \epsilon(l', W_1^{sp}) = l + \epsilon\frac{\partial W_1^{sp}}{\partial L'} \tag{42}$$

The long period to osculating elements transformations are then given by

$$L = L' - \frac{\epsilon}{4L'^3}\left[\left(-1+3\frac{H'^2}{G'^2}\right)\left(\frac{a'^3}{r'^3}-\frac{L'^3}{G'^3}\right)+3\left(1-\frac{H'^2}{G'^2}\right)\left(\frac{a'}{r'}\right)^3\cos(2f'+2g')\right] \tag{43a}$$

$$G = G' - \frac{3\epsilon}{4G'^3}\left(1-\frac{H'^2}{G'^2}\right)\left[\cos(2f'+2g')+e'\cos(f'+2g')+\frac{e'}{3}\cos(3f'+2g')\right] \tag{43b}$$

$$H = H' \tag{43c}$$

$$\begin{aligned}
l = l' &+ \frac{\epsilon}{8e'L'^4}\left(\frac{L'}{G'}\right)\left[2\left(-1+3\frac{H'^2}{G'^2}\right)\left(\frac{a'^2}{r'^2}\frac{G'^2}{L'^2}+\frac{a'}{r'}+1\right)\sin f'\right. \\
&+3\left(1-\frac{H'^2}{G'^2}\right)\left(\left(-\frac{a'^2}{r'^2}\frac{G'^2}{L'^2}-\frac{a'}{r'}+1\right)\sin(f'+2g')\right. \\
&+\left.\left.\left(\frac{a'^2}{r'^2}\frac{G'^2}{L'^2}+\frac{a'}{r'}+\frac{1}{3}\right)\sin(3f'+2g')\right)\right]
\end{aligned} \tag{43d}$$

$$\begin{aligned}
g = g' &- \frac{\epsilon}{8e'L'^4}\left(\frac{L'}{G'}\right)^2\left[2\left(-1+3\frac{H'^2}{G'^2}\right)\left(\frac{a'^2}{r'^2}\frac{G'^2}{L'^2}+\frac{a'}{r'}+1\right)\sin f'\right. \\
&+3\left(1-\frac{H'^2}{G'^2}\right)\left(\left(-\frac{a'^2}{r'^2}\frac{G'^2}{L'^2}-\frac{a'}{r'}+1\right)\sin(f'+2g')\right. \\
&+\left.\left.\left(\frac{a'^2}{r'^2}\frac{G'^2}{L'^2}+\frac{a'}{r'}+\frac{1}{3}\right)\sin(3f'+2g')\right)\right] \\
&-\frac{3\epsilon}{8G'^4}\left[2\left(-1+5\frac{H'^2}{G'^2}\right)(f'-l'+e'\sin f')\right. \\
&+\left.\left(3-5\frac{H'^2}{G'^2}\right)\left(\sin(2f'+2g')+e'\sin(f'+2g')+\frac{e'}{3}\sin(3f'+2g')\right)\right]
\end{aligned} \tag{43e}$$

$$\begin{aligned}
h = h' &+ \frac{3\epsilon}{4G'^4}\frac{H'}{G'}\left(2(f'-l'+e'\sin f')-\sin(2f'+2g')-e'\sin(f'+2g')\right. \\
&-\left.\frac{e'}{3}\sin(3f'+2g')\right)
\end{aligned} \tag{43f}$$

The transformation between long period to mean elements is achieved in a similar manner. For the (L', l') elements, they are given by

$$L' = L'' + \epsilon(L'', W_1^{lp}) = L'' - \epsilon\frac{\partial W_1^{lp}}{l''} \tag{44}$$

$$l' = l'' + \epsilon(l'', W_1^{lp}) = l'' + \epsilon\frac{\partial W_1^{lp}}{L''} \tag{45}$$

with the remaining transformations generated in an analogous manner. The inverse transformation is again achieved by simply switching the prime's and double-prime's and reversing the sign of the ϵ term. The transformation from mean to long period elements is given by

$$L' = L'' \tag{46a}$$

$$G' = G'' + \frac{\epsilon}{16G''^3}\left(1-\frac{G''^2}{L''^2}\right)\left(1-16\frac{H''^2}{G''^2}+15\frac{H''^4}{G''^4}\right)\left(1-5\frac{H''^2}{G''^2}\right)^{-1}\cos 2g'' \tag{46b}$$

$$H' = H'' \tag{46c}$$

$$l' = l'' - \frac{\epsilon}{16G''^4}\left(\frac{G''^3}{L''^3}\right)\left(1-16\frac{H''^2}{G''^2}+15\frac{H''^4}{G''^4}\right)\left(1-5\frac{H''^2}{G''^2}\right)^{-1}\sin 2g'' \tag{46d}$$

$$g' = g'' + \frac{\epsilon}{32G''^4}\left(1 - 5\frac{H''^2}{G''^2}\right)^{-1}\left[\left(3 - \frac{G''^2}{L''^2}\right)\left(1 - 16\frac{H''^2}{G''^2} + 15\frac{H''^4}{G''^4}\right)\right.$$

$$\left. - 2\frac{H''^2}{G''^2}\left(1 - \frac{G''^2}{L''^2}\right)\left(11 + 25\frac{H''^2}{G''^2} + 200\frac{H''^4}{G''^4}\left(1 - 5\frac{H''^2}{G''^2}\right)^{-1}\right)\right]\sin 2g'' \tag{46e}$$

$$h' = h'' + \frac{\epsilon}{16G''^4}\left(\frac{H''}{G''}\right)\left(1 - \frac{G''^2}{L''^2}\right)\left(1 - 5\frac{H''^2}{G''^2}\right)^{-1}$$

$$\left(11 + 25\frac{H''^2}{G''^2} + 200\frac{H''^4}{G''^4}\left(1 - 5\frac{H''^2}{G''^2}\right)^{-1}\right)\sin 2g'' \tag{46f}$$

PERFORMANCE EVALUATION OF THE GPS ONBOARD ORBIT DETERMINATION SOFTWARE TO ENSURE IMPROVED POSITIONING ACCURACY

Andrew R. Garber, Lin Haas, Mark Pittelkau
Orbital Sciences Corporation

ABSTRACT

This paper presents the performance validation of the GPS Onboard Orbit Determination Software (GOODS). Orbital developed GOODS in response to the need for improving GPS real-time positioning accuracies to better than 20 m 1σ. The GOODS software includes an Extended Kalman Filter that processes GPS pseudorange and Doppler measurements. Contained within the filtering algorithms are high fidelity models of aerodynamic drag, Solar/Lunar gravity, and non-spherical geopotential. GOODS will fly on Orbview-3, Orbview-4, VCL, and RADARSAT-2.

Analysis focuses on the validation of individual force models and state estimation performance. Specific test cases evaluate a reduced order of the geopotential field, orbit adjust maneuvers, and a 7 day extended duration processing interval. Results show GOODS meets the success criteria for all of eleven test cases. Position and velocity errors obtained through Monte Carlo runs are consistently less than 10.0 m and 0.01 m/s 1σ, respectively.

INTRODUCTION

The GOODS software was developed to provide high-accuracy real-time positioning capabilities for satellites using space-capable GPS receivers. Existing commercial space-capable receivers (single frequency-SPS) typically provide positioning errors on the order of 100 m 95% and velocity errors of 1 m/s 95%. Orbital is currently building several satellites that require real-time 1σ position and velocity errors on the order of 20 m and 0.01 m/s, respectively.

To achieve these accuracies, GOODS uses an Extended Kalman filter (EKF) with high-fidelity force models. These force models include the JGM-2 geopotential (30th order), the Harris-Priester atmospheric density model for drag acceleration, and Solar/Lunar point mass accelerations. A 4th order Runge-Kutta numerical integrator is used to propagate the state vector between epochs. The filter itself uses a 9-element state vector, that includes position, velocity, drag coefficient, and receiver clock bias and drift. State process noise for position and velocity are based on errors of omission for the non-spherical geopotential model, along with other empirical noise factors. The receiver clock and drag coefficient are modeled as random walk processes.

Eleven test cases were developed to validate the GOODS performance. Table 1 provides an overview of the test cases, along with the success criteria for each case. The success criteria are based on existing Orbital spacecraft positioning accuracy requirements. The test cases were designed to isolate specific GOODS capabilities, and were executed such that each successive test built upon prior successful test results. The start time for each test case is midnight, 08 September 1998. The duration of each test was 24 hours except for test case 10 that has a 7 day duration.

Test cases 1-2 validate the GOODS propagation force models. A reference trajectory generated by the MicroCosm program (ref. 1) was used as the truth trajectory for these tests. Test cases 3-10 validate the GOODS filter performance. To provide a controlled test environment for evaluating the GOODS filter, a GPS simulator was developed for generating pseudorange measurements from a reference trajectory. The simulator was designed to provide control over the individual error components of the pseudorange measurements, including receiver clock bias (C_{user}), satellite clock bias (C_{GPS}), Selective Availability (SA), ionospheric delay (I), receiver noise (η_{user}), and User Equivalent Ranging Error (UERE). Finally, test case 11 performs a Monte Carlo analysis to evaluate the statistical performance of GOODS.

Table 1: GOODS Evaluation Test Matrix

#	Title	Description	Success Criteria
1	Propagation (Partial Models)	Validate the non-drag force perturbation models over a 24 hour period. Perfect a priori position and velocity.	• max position error < 10 m per component
2	Propagation (Full Models)	Validate drag model over a 24 hour period. Perfect a priori position and velocity.	• max position error < 200 m per component
3	Baseline Filter	Validate basic filter performance given perfect observations and a perfect a priori state.	• 1σ position error < 10 m • 1σ velocity error < 0.01 m/s • filter converges within 2 orbits
4	Solve for Position/Velocity	Evaluate the ability to solve for position and velocity given that the a priori position and velocity are offset by 3σ receiver point solution errors. Perfect a priori clock and C_D.	• 1σ position error < 20 m • 1σ velocity error < 0.01 m/s • filter converges within 2 orbits
5	Solve for Clock	Evaluate the ability to solve for a clock bias and drift given that the a priori clock bias and drift are offset by 3σ receiver point solution errors. Perfect a priori position, velocity, and C_D.	• 1σ position error < 20 m • 1σ velocity error < 0.01 m/s • filter converges within 2 orbits
6	Solve for Drag	Evaluate the ability to solve for drag given that the a priori C_D is offset by 50%. Perfect a priori position, velocity, and clock.	• 1σ position error < 20 m • 1σ velocity error < 0.01 m/s • C_D converges within 10 orbits
7	Third-Body Gravity	Evaluate the state propagation errors without the presence of the Solar/Lunar perturbations. Perfect a priori states. If conditions are not met, include only solar gravity. Remaining cases will use the 3rd body gravitation selection determined in this case.	24 Hr Propagation RIC Errors: • 1σ radial position error < 100 m • 1σ intrack position error < 500 m • 1σ crosstrack position error < 100 m
8	Geopotential Model	Step 1: Evaluate propagation errors errors as a function of the gravity model order. Starting with order 20, decrease the order by half in successive runs until the success criteria is not met. Then increase the order until the success criteria is met. Remaining test cases will use the gravity order determined in this case. Step 2: Evaluate state estimation errors with determined gravity order.	24 Hr Propagation Errors: • 1σ intrack position error < 500 m 24 Hour Estimation Errors: • 1σ position error < 20 m • 1σ velocity error < 0.01 m/s
9	Solve for State	Evaluate the ability to solve for all state parameters with the gravity conditions determined in Cases 7 and 8. A priori state offset by the 3σ Viceroy values (C_D offset by 50%).	• 1σ position error < 20 m • 1σ velocity error < 0.01 m/s • C_D converges within 10 orbits
10	Extended Duration with Maneuvers	Evaluate the ability to model maneuver uncertainties. Measurements are processed during maneuvers. Evaluate the state estimation process over an extended period of 7 days.	• 1σ position error < 20 m • 1σ velocity error < 0.01 m/s • C_D converges to nominal value • Covariance increases during maneuver • Solution remains converged (except possibly for a short time after the maneuvers)
11	Monte Carlo Analysis	Validate the performance of the GOODS state estimation process by combining results of 30 cycles, each with a different initial state vector offset with random values determined from the error distribution of a GPS receiver state vector. Evaluate the statistical performance of GOODS.	• 1σ position error < 20 m • 1σ velocity error < 0.01 m/s • C_D converges within 10 orbits

GOODS VALIDATION

Test Case 1 – Propagation Using Partial Force Models

Test case 1 validates the GOODS propagator, the geopotential, and Solar/Lunar force models. Drag is not modeled in this test case. GOODS force models include the JGM-2 geopotential (30th order), and Solar/Lunar point mass accelerations. A 4th order Runge-Kutta numerical integrator is used to propagate the GOODS state vector between epochs. The MicroCosm truth trajectory was generated using a 30th order GEM-T3 geopotential model, and the JPL planetary ephemeris for Solar/Lunar perturbations. MicroCosm uses a more accurate Cowell predictor/corrector numerical integrator. The accuracy of the GOODS propagated state vector relative to MicroCosm is evaluated over a 23 hour timespan. Table 4 shows that the test criteria of maximum position errors < 10 m are met.

Figure 1 shows the radial/intrack/crosstrack GOODS position errors relative to the truth trajectory. The small errors in each component validate the GOODS partial force models, and present their high level of accuracy.

Table 4: Test Case 1 Propagation Errors with Partial Models

Position Errors			Velocity Errors		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	-0.03	0.44	Radial Error	0.001	0.009
Intrack Error	-0.91	2.97	Intrack Error	0.000	0.001
Crosstrack Error	0.03	1.00	Crosstrack Error	0.000	0.008

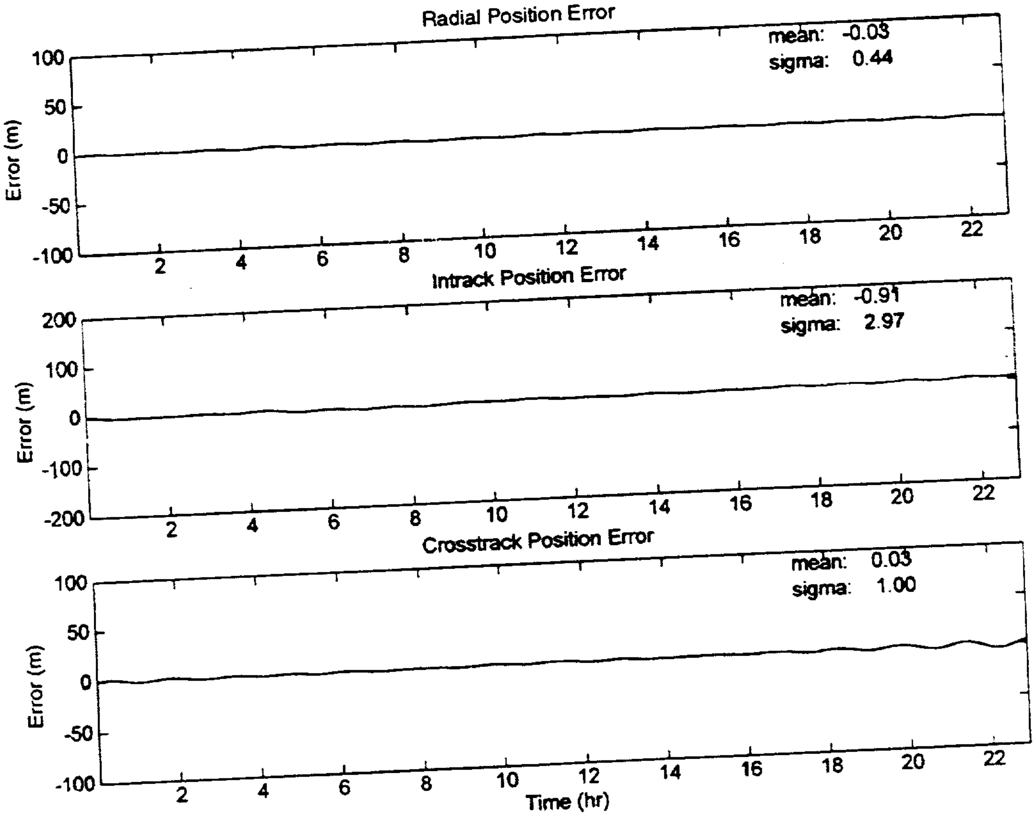


Figure 1: Test Case 1 GOODS Position Errors Relative to the MicroCosm Truth Trajectory

Test Case 2 – Propagation Using Full Force Models

Test Case 2 validates the GOODS drag model. In computing the drag acceleration, GOODS employs the Harris-Priester atmospheric density model, whereas MicroCosm uses the higher fidelity 1971 Jacchia-Roberts atmospheric density model. The cross-sectional area of the satellite was 4.2 m^2.

Initially for both drag coefficients (C_D) equal to 2.3, over the 23 hour timespan, the propagation of the initial state vector takes approximately 10 hours to diverge to -200 m in intrack position error. This is due to the different atmospheric drag models. The mean radial and crosstrack errors maintain good stability at about 0.0 m mean error. To account for the differences between the MicroCosm and GOODS drag models, the C_D was set to 2.433 in GOODS. The value of 2.433 was chosen to minimize the intrack position errors over 23 hours.

Figure 2 shows the radial/intrack/crosstrack GOODS position errors relative to the truth trajectory. Over the 23 hour timespan, the maximum intrack position error is -85 m. In setting C_D to 2.433, control of such intrack errors is obtained, and the precision of the GOODS drag model is validated. During real-time operations, GOODS will estimate C_D to compensate for errors in the drag model. From Table 5 and Figure 2, the test criteria of maximum position errors < 200 m are met.

Table 5: Test Case 2 Propagation Errors with Full Models

Position Errors			**Velocity Errors**		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	-0.06	8.58	Radial Error	0.034	0.031
Intrack Error	-27.63	28.54	Intrack Error	0.000	0.010
Crosstrack Error	0.02	0.94	Crosstrack Error	0.000	0.008

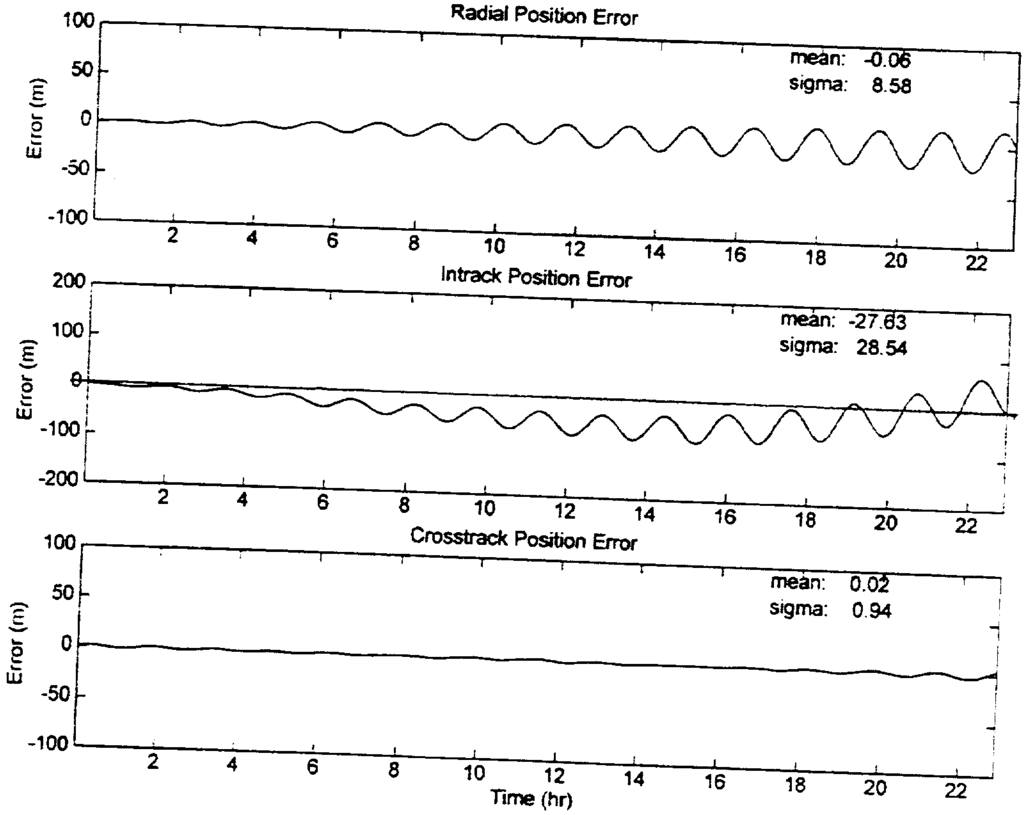


Figure 2: Test Case 2 GOODS Position Errors Relative to the MicroCosm Truth Trajectory for C_D = 2.433

Test Case 3 – Solve for Baseline

Test Case 3 validates the fundamental Kalman filter equations and logic. In this case, perfect pseudorange measurements are simulated using zero measurement errors in the SA, receiver noise, receiver clock bias, and receiver clock drift. The initial state vector consists of perfect *a priori* position, velocity, clock bias, clock drift, and drag coefficient obtained from Case 2 (C_D = 2.433). From Table 6, the following test criteria are met: 1) 1σ maximum position errors < 10 m, and 2) 1σ maximum velocity errors < 0.01 m/s. In addition, the filter converges well within 2 orbits.

Table 6: Baseline State Errors

Position Errors			Velocity Errors		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	-0.51	0.98	Radial Error	-0.001	0.011
Intrack Error	0.84	1.45	Intrack Error	0.000	0.001
Crosstrack Error	-0.03	0.44	Crosstrack Error	0.000	0.009

Test Case 4 – Solve for Position and Velocity

Test Case 4 validates the ability of the Kalman filter to solve for position and velocity using non-perfect pseudorange measurements and non-perfect *a priori* position and velocity. Pseudorange measurement errors include SA, and receiver noise. Receiver clock errors are not modeled. The initial state vector is perturbed by 3σ position and velocity errors of a typical GPS space receiver, and uses perfect *a priori* clock and C_D. Initial 3σ position errors are 450 m, 300 m, and 300 m in radial, intrack, and crosstrack components, respectively. Initial 3σ velocity errors are 5.196 m/s in each of the three components. The initial position and velocity variances were modified to account for the initial position and velocity errors. From Table 7, the following test criteria are met: 1) 1σ maximum position errors < 20 m, and 2) 1σ maximum velocity errors < 0.01 m/s. In addition, the filter converges well within 2 orbits.

Table 7: State Errors Using Initial 3σ Position and Velocity Offsets

Position Errors			Velocity Errors		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	-0.48	2.42	Radial Error	0.000	0.014
Intrack Error	-0.07	6.20	Intrack Error	0.000	0.003
Crosstrack Error	-0.77	6.21	Crosstrack Error	0.000	0.010

Test Case 5 – Solve for Clock Bias and Drift

Test Case 5 validates the ability of the Kalman filter to solve for receiver clock parameters using non-perfect pseudorange measurements and non-perfect *a priori* clock bias and clock drift. Pseudorange measurement errors include 23.0 m (1σ) for SA, 3.60 m (1σ) for receiver noise, 150000 m for initial receiver clock bias, and –0.30 m/s for initial receiver clock drift. The initial state vector uses perfect *a priori* position, velocity, and C_D. Initial state clock errors are 450000 m and –0.9 m/s for the clock bias and the clock drift, respectively. Initial receiver clock bias offset and clock drift variances were modified to account for initial state clock errors. From Table 8, the following test criteria are met: 1) 1σ maximum position errors < 20 m, and 2) 1σ maximum velocity errors < 0.01 m/s. In addition, the filter converges well within 2 orbits.

Table 8: State Errors Using Initial Clock Bias and Drift Rate Offsets

Position Errors			Velocity Errors		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	-0.22	2.04	Radial Error	0.001	0.012
Intrack Error	-0.44	5.46	Intrack Error	0.000	0.002
Crosstrack Error	0.30	6.58	Crosstrack Error	0.000	0.009

Test Case 6 – Solve for Drag

Test Case 6 validates the ability of the Kalman filter to solve for drag using non-perfect pseudorange measurements and a non-perfect *a priori* coefficient of drag. Pseudorange measurement errors include SA and receiver noise. Receiver clock errors are not modeled. The initial state vector consists of a non-perfect *a priori* drag coefficient offset 50% from the C_D of 2.433, with perfect *a priori* position, velocity, and clock. The initial atmospheric drag coefficient variance was modified to account for the initial coefficient's error. From Table 9, the following test criteria are met: 1) 1σ maximum position errors < 20 m, and 2) 1σ maximum velocity errors < 0.01 m/s. In addition, the coefficient of drag converges to 2.433 in less than 10 orbits.

Table 9: State Errors Using Initial C_D Offset 50 % from True Value

Position Errors			**Velocity Errors**		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	-1.09	2.75	Radial Error	-0.002	0.016
Intrack Error	1.30	7.46	Intrack Error	0.001	0.003
Crosstrack Error	0.13	7.45	Crosstrack Error	0.000	0.009

Test Case 7 – Third-Body Gravity Perturbations of the Sun and Moon

Test Case 7 verifies the necessity of modeling Solar/Lunar perturbations, as several Orbital spacecraft have stringent memory and processing specifications that may require this GOODS capability to be disabled. To observe third body effects on the propagation accuracy, GOODS was run first with Solar/Lunar perturbations on and then with them off. Measurement processing was inhibited in both cases. This was done for both a 450 km and an 1000 km orbit. This test uses a perfect *a priori* position, velocity, clock bias, clock bias drift, and coefficient of drag.

At the altitude of 450 km, the coefficient of drag was set to 2.433 for propagating the initial state vector. Over 23 hours, residuals between Solar/Lunar "On" state errors vs. Solar/Lunar "Off" state errors show good agreement in radial and crosstrack components. Intrack position errors grow to 120 m in the 23 hour period, as shown in Figure 3.

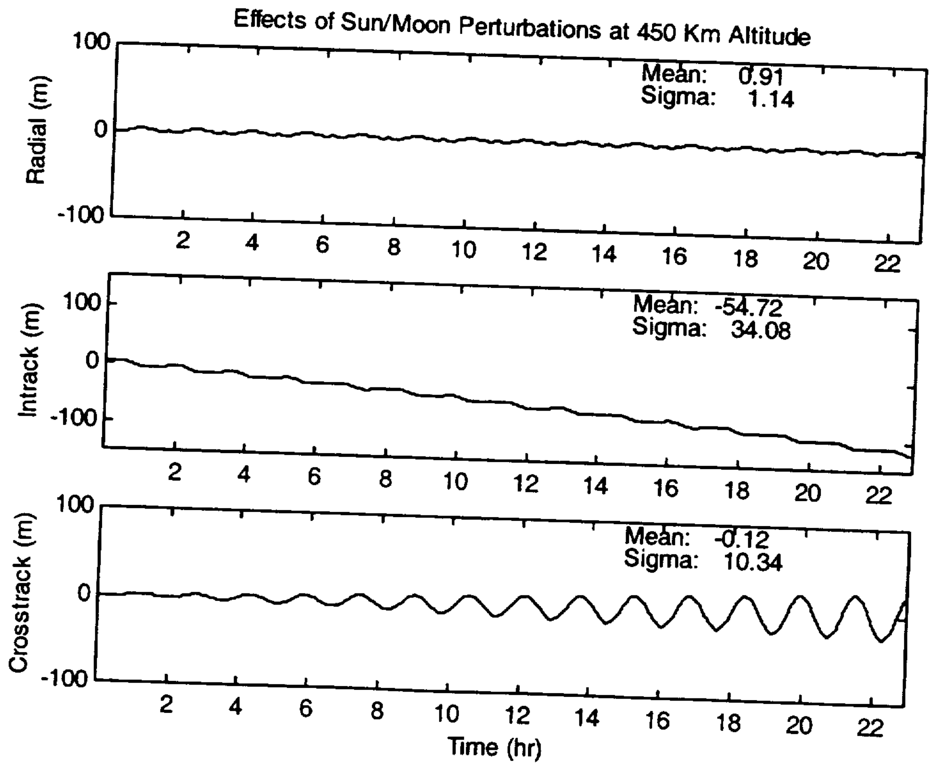


Figure 3: Third Body Propagation Errors at Altitude of 450 Km

At an altitude of 1000 km, the coefficient of drag was set to $C_D = 0.0078$ for propagating the initial state vector. In doing so, effects of Solar/Lunar perturbations are isolated at the higher altitude. Over 23 hours, residuals between state vectors show good agreement in radial and crosstrack components, as before. Intrack position errors grow to 140 m in a 23 hour timespan. The purpose of testing at an altitude of 1000 km is due to the prevalent effects of Solar/Lunar accelerations at higher altitudes. In addition, 1000 km is the maximum altitude for Orbital satellites that shall employ GOODS.

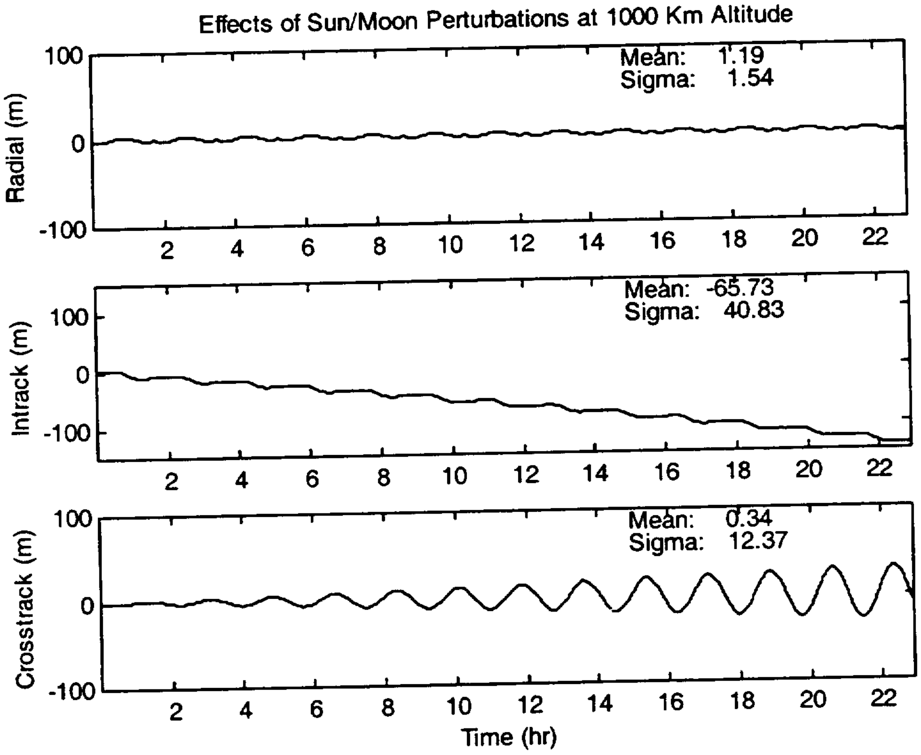


Figure 4: Third Body Propagation Errors at Altitude of 1000 Km

For both tests, the following success criteria are met: 1) 1σ radial position error < 100 m, 2) 1σ intrack position error < 500 m, and 3) 1σ crosstrack error < 100 m.

Since the test criteria are met, all remaining test cases exclude Solar/Lunar perturbations from processing.

Test Case 8 – Geopotential Model

Test Case 8 determines the minimum order of the geopotential model such that GOODS meets both the 23 hour 1σ propagation requirement of < 500 m intrack error, and the 1σ estimation position and velocity accuracy requirements of 20 m, and 0.01 m/s, respectively. As in the previous case, stringent memory and processing requirements of several Orbital spacecraft may require GOODS to truncate the geopotential model. This was achieved in two steps. The first step involved propagating the perfect a priori state vector with decreasing field order, until the success criteria was not met. The 17th order geopotential field model permitted propagated intrack errors < 500 m over the 23 hour timespan.

The second step validated the selection of 17th order field model by evaluating the estimation errors using non-perfect pseudorange observations and perfect *a priori* state. Pseudorange measurement errors include SA and receiver noise. The initial state vector consists of perfect *a priori* position, velocity, clock bias, clock drift, and drag coefficient. The state process noise was modified to account for the lower-order geopotential, based upon propagation results using the 17th order field model. Solar/Lunar perturbations are excluded from this processing.

For the 23 hour timespan, Table 10 shows the estimation errors for processing with the 17th order geopotential field model. The following test criteria are met: 1) 1σ maximum position errors < 20 m, and 2) 1σ maximum velocity errors < 0.01 m/s.

Table 10: State Errors Solving with 17th Order Geopotential Field

Position Errors			**Velocity Errors**		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	0.11	4.59	Radial Error	0.001	0.016
Intrack Error	-0.68	7.69	Intrack Error	0.000	0.005
Crosstrack Error	-0.64	6.17	Crosstrack Error	0.001	0.010

For missions that employ the GOODS software, the selected geopotential field order depends on the positioning accuracy requirement. In order to meet the 20 m (1σ) GOODS test requirement, the order must be $\geq$ 17. As a result, test cases 9 and 10 shall use the 17th order geopotential.

Test Case 9 – Solve for State

Test Case 9 validates the ability of the Kalman filter to solve for the entire state vector using non-perfect pseudorange measurements and non-perfect *a priori* state vector. Pseudorange measurement errors include SA, receiver noise, receiver clock bias, and receiver clock drift. The initial state vector is perturbed by 3σ position and velocity errors, clock bias, and clock drift errors of a typical GPS space receiver. The *a priori* C_D is offset by 50% from the true value of 2.433. Initial variances for position, velocity, coefficient of drag, and receiver clock were modified accordingly, along with radial, intrack, and crosstrack state process noise based upon the 17th order geopotential. Solar/Lunar perturbations are excluded from processing in the filtering algorithms.

From Table 11, for the 23 hour timespan, the following test criteria are met: 1) 1σ maximum position errors < 20 m, and 2) 1σ maximum velocity errors < 0.01 m/s. In addition, the C_D converges to the true value of 2.433 in under 10 orbits.

Table 11: State Errors Solving for Entire State Vector

Position Errors			**Velocity Errors**		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	-1.44	5.29	Radial Error	-0.002	0.016
Intrack Error	1.04	8.21	Intrack Error	0.001	0.005
Crosstrack Error	0.37	6.67	Crosstrack Error	0.000	0.009

Test Case 10 – Extended Duration with Maneuvers

Test Case 10 validates GOODS over an extended duration of 1 week, and includes 3 orbit adjust maneuvers. The maneuvers occur at 40 hour intervals, each with a 120 second duration. In order, an intrack maneuver occurs at 980910.1600 UTC, a crosstrack maneuver occurs at 980912.0800 UTC, and a radial maneuver occurs at 980914.0000 UTC. The ΔV for the intrack, crosstrack and radial maneuvers are 1.562 m/s, 3.104 m/s, and 2.492 m/s, respectively. The intrack and crosstrack maneuvers were modeled after TDRSS maneuvers, but scaled down for the VCL spacecraft weight. The radial maneuver is based upon the orbit adjust maneuvers of UARS and GRO.

Non-perfect pseudorange measurement errors include SA, receiver noise, receiver clock bias, and receiver clock drift. The initial state vector is perturbed by 3σ position and velocity errors, clock bias, and clock drift errors of a typical GPS space receiver. The *a priori* C_D is offset by 50% from the true value of 2.433.

GOODS accounts for maneuver effects in the state propagation by instantaneously 'bumping' the state process noise at the beginning of the maneuver. The magnitude of the increase is a function of the total ΔV of the maneuvers. Maneuver parameters are input to GOODS via an uplink command dataset. This occurs at the onset of GOODS execution, as each maneuver is stored in a queue and is executed at the proper start time. Maneuver parameters include start and stop times, and position and velocity process noise based on the maneuver magnitude and direction. In addition, the state process noise was modified to account for the lower-order geopotential, based upon propagation results using the 17th order field model. Other initial variances modified were those for position, velocity, coefficient of drag, and receiver clock. Solar/Lunar perturbations are excluded from processing in the filtering algorithms.

For the 1 week timespan, the following test criteria are met: 1) 1σ maximum position errors < 20 m, and 2) 1σ maximum velocity errors < 0.01 m/s. In addition, the C_D converges to its true value of 2.433 after the first two maneuvers (intrack and crosstrack), but takes longer to reconverge after the radial maneuver, as shown in Figure 7. This characteristic is a function of maneuver magnitude, position and velocity correlations, filter gain, etc.

From Table 12, residuals between the GOODS filtered state vector and the truth trajectory show good agreement. The only exceptions occur during times of maneuvers. During the maneuvers, the maneuver process noise increases the filter covariance that allows the estimated state to be more sensitive to the incoming measurements. This results in larger state errors during the maneuver. As more measurements are processed, the filter covariance decreases and the state errors reconverge to the pre-maneuver levels. Figures 5 and 6 present filtered state errors for position and velocity over the entire 1 week timespan. The ± 3σ values contained within Figures 5 and 6 are based on the filter variances. Figure 7 shows filtered state vector parameter C_D, along with the true $C_D = 2.433$, depicted as a solid line.

Table 12: State Errors Solving for Maneuvers and Extended Duration

Position Errors			**Velocity Errors**		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial Error	-0.60	10.81	Radial Error	-0.002	0.031
Intrack Error	0.89	9.89	Intrack Error	0.000	0.017
Crosstrack Error	-0.25	8.17	Crosstrack Error	0.000	0.020

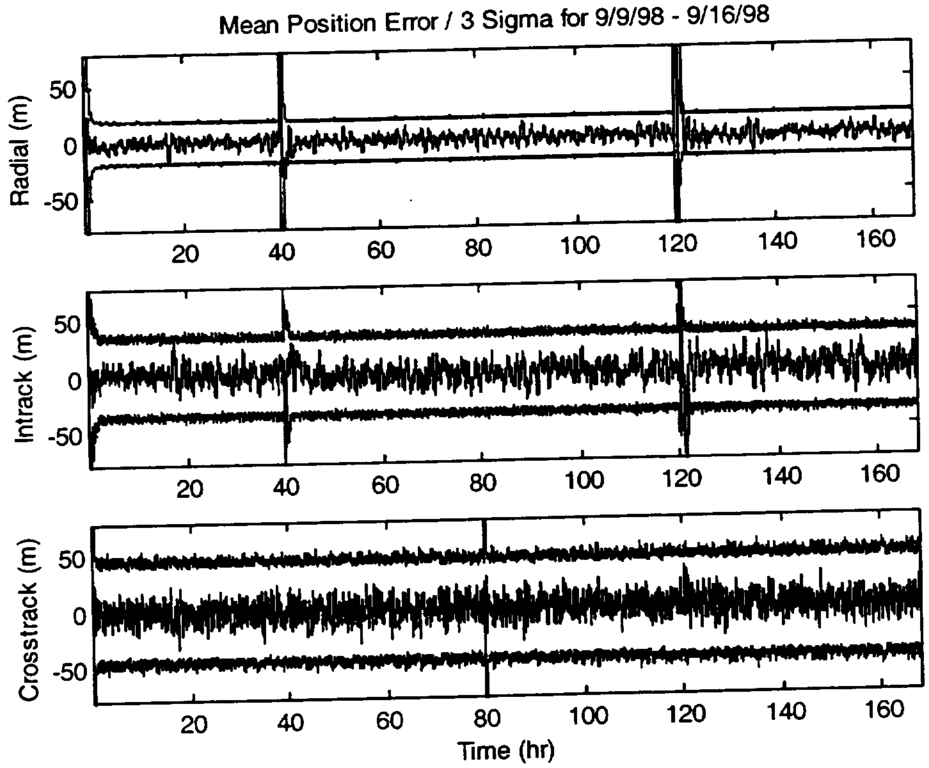


Figure 5: Filtered Position Errors for Extended Duration Plus Maneuvers

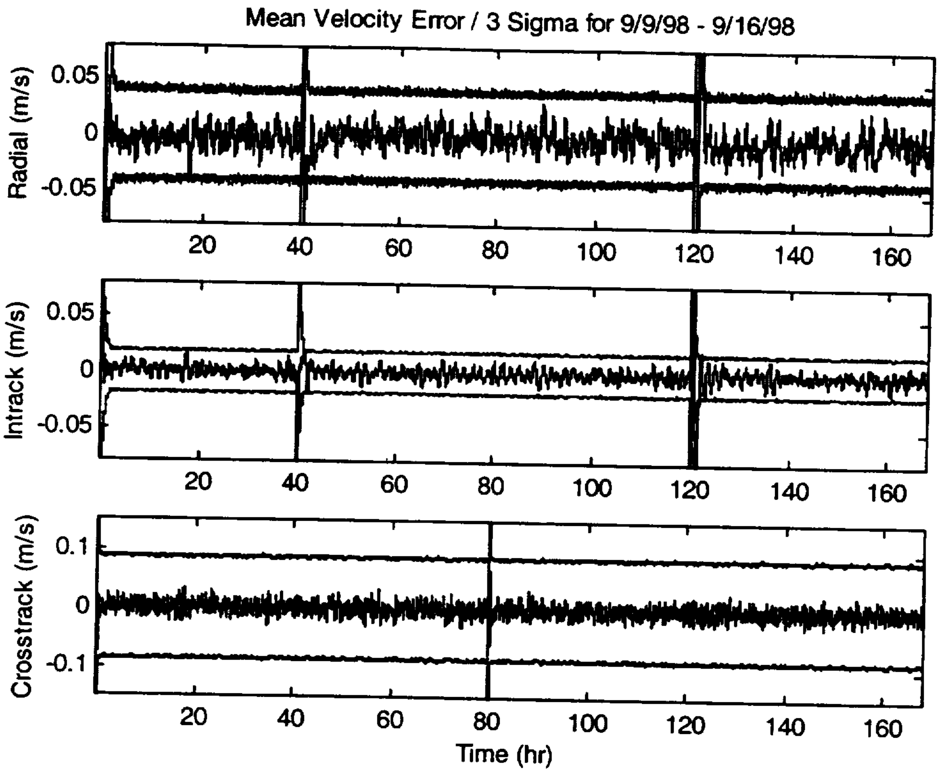


Figure 6: Filtered Velocity Errors for Extended Duration Plus Maneuvers

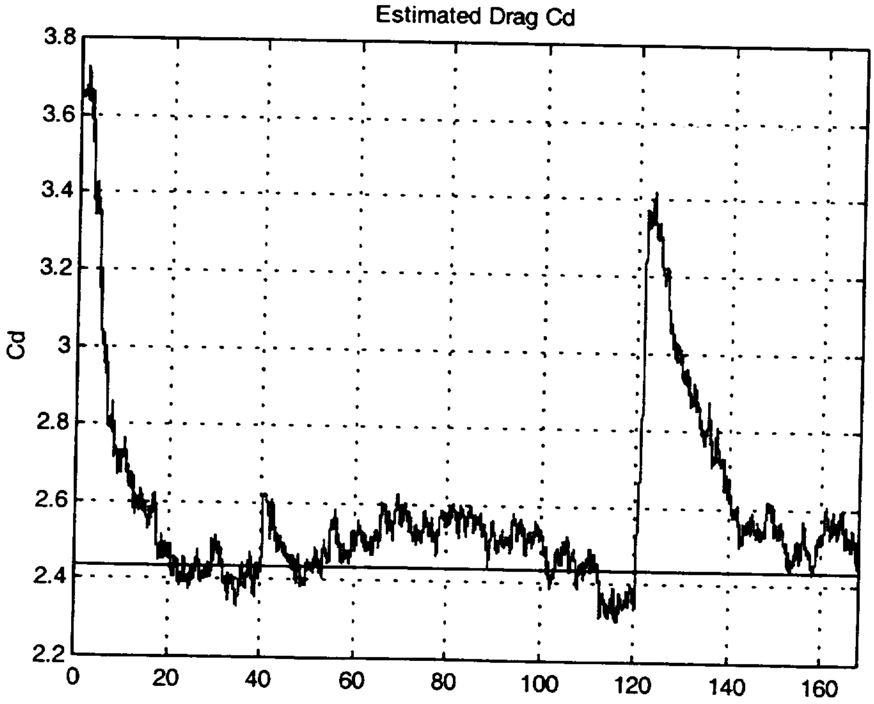


Figure 7: Filtered C_D for Extended Duration Plus Maneuvers

Test Case 11 – Monte Carlo Analysis

Test case 11 evaluates the statistical behavior of GOODS by using a Monte Carlo technique. The GPS simulator is configured to generate 30 cycles of measurement data, each cycle containing statistically independent values for the receiver clock, SA, receiver noise, and UERE errors applied to the pseudorange measurement. Each GOODS cycle starts with a different initial state vector, whose value is determined by perturbing the true initial state with random values determined from the error distribution of a GPS receiver state vector. The mean and standard deviation of the position and velocity errors are then computed across the 30 cycles, and represent the performance of the GOODS software under a multitude of different conditions for the given orbit. For this analysis, GOODS uses the 30th order geopotential.

The modeled equation for pseudorange is:

$$P_r = R + I + C_{user} - C_{GPS} + SA + \eta_{user} + UERE$$

where:

P_r . pseudorange measurement
R - geometric slant range
I - Ionospheric delay
C_{user} - User Receiver clock bias
C_{GPS} - GPS satellite clock bias (including relativistic effects)
SA - Selective Availability
η_{user} - User Receiver noise
UERE - User Equivalent Range Error to simulate errors in the broadcast ephemeris and clock parameters

Ionospheric delay is modeled as a function of elevation and Total Electron Content (TEC). The user clock and UERE are both modeled as random walk processes. The user clock model is specified by the Allen variances for a Temperature Compensated Crystal Oscillator (TCXO). SA is modeled as a 2nd order Gauss-Markov process (ref. 2). In GOODS, I, SA, η_{user}, and UERE are unmodeled error sources in the Kalman filtering algorithms. They are accounted for in the measurement weights. SA is a time correlated random process, whereas η_{user} is a pure random process. R is the magnitude of the difference between the position of the GPS satellite and that of the user. UERE compensates for the random time correlated errors in the broadcast ephemeris and GPS satellite clock errors. C_{GPS} is the random walk defined by the TCXO Allen variances.

The Monte Carlo analysis was performed over 24 hours with no maneuvers. Table 13 shows the average mean and standard deviation statistics computed over the 30 Monte Carlo cycles between 2 and 24 hours (filter converged). The -2.58 m bias in the mean intrack error is caused by the non-zero mean ionospheric errors. Both the position and velocity meet the 1σ performance requirements of 20 m, and 0.01 m/s, respectively.

Figure 8 presents the mean and ± 3σ position errors for the 30 Monte Carlo cycles. These mean errors remain at about 0.0 m for most of the 24 hour timespan, with the exception at the onset of filtering from t = 0 to t ≈ 2.5 hours. Figure 9 presents an overlay of the position errors for each of the 30 cycles. The jaggedness for each position component is a result of varying initial state vectors with random values determined from the error distribution of a GPS receiver derived state vector.

Figure 10 presents the mean and ± 3σ velocity errors for the 30 Monte Carlo cycles. These mean errors remain at about 0.0 m/s for most of the 24 hour timespan, with the exception at the onset of filtering from t = 0 to t ≈ 2.0 hours. Figure 11 presents an overlay of the velocity errors for each of the 30 cycles. The jaggedness for each velocity component is a result of varying initial state vectors with random values determined from the error distribution of a GPS receiver derived state vector.

Table 13: Monte Carlo Errors for 980909.0000Z – 980910.0000Z With No Maneuvers

Position Errors			**Velocity Errors**		
Component	**Mean (m)**	**Sigma (m)**	**Component**	**Mean (m/s)**	**Sigma (m/s)**
Radial	0.30	1.45	Radial	0.003	0.005
Intrack	-2.58	5.05	Intrack	0.000	0.001
Crosstrack	0.03	2.71	Crosstrack	0.000	0.003

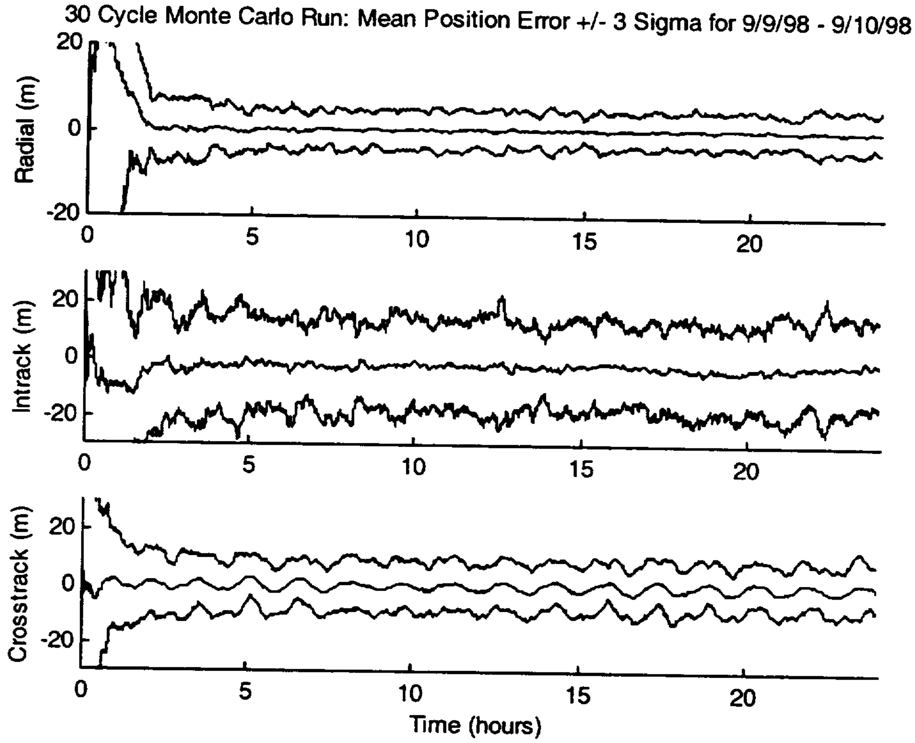


Figure 8: Position Errors for Simulated Data with No Maneuvers

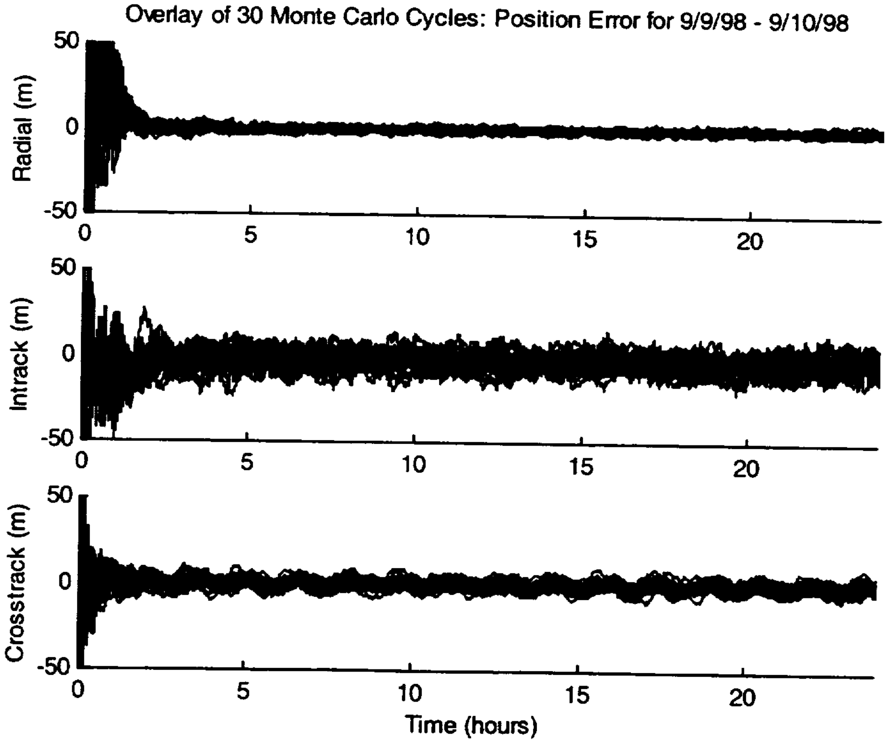


Figure 9: Overlay of Position Errors for Each of 30 Monte Carlo Cycles

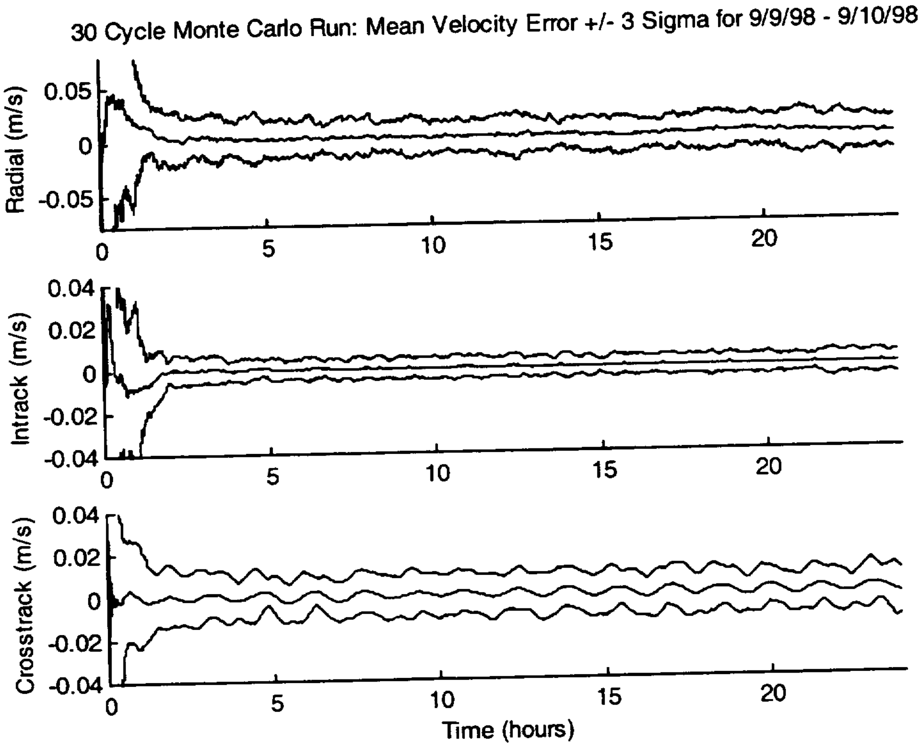


Figure 10: Velocity Errors for Simulated Data with No Maneuvers

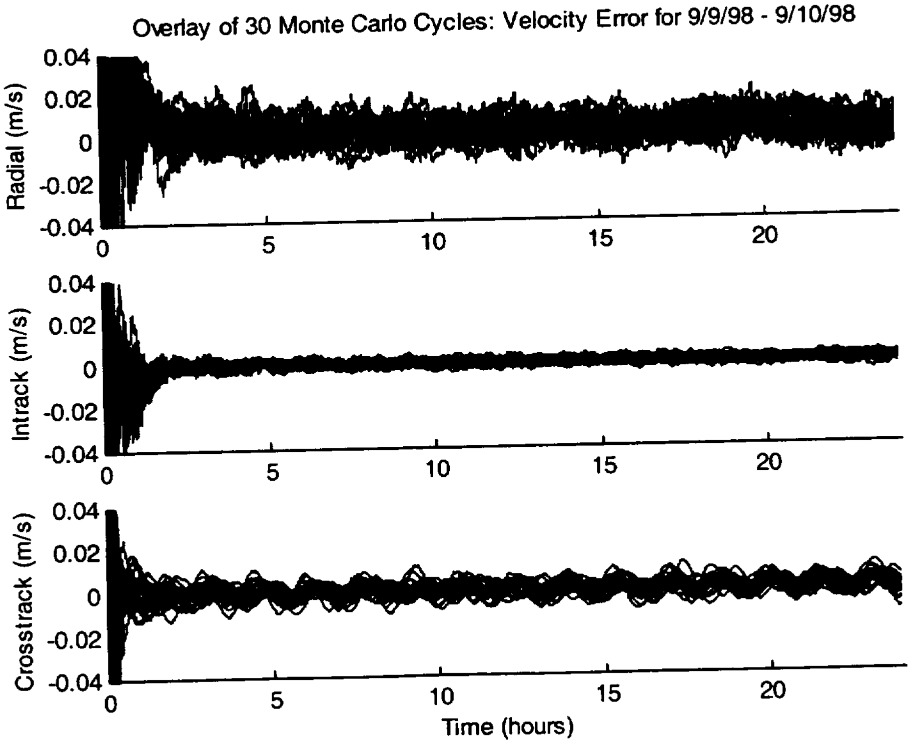


Figure 11: Overlay of Velocity Errors for Each of 30 Monte Carlo Cycles

CONCLUSIONS

An incremental approach was presented for validating the GOODS force models and estimation capabilities. The results presented show that GOODS meets the performance requirements for Orbital spacecraft, which are 1σ position and velocity errors less than 20 m and 0.01 m/s, respectively. Test cases included worst-case initial state errors (based on 3σ errors from a GPS receiver), orbit adjust maneuvers, and a 7 day extended duration run.

A Monte Carlo analysis was performed to quantify the statistical behavior of GOODS for a given orbit and environment. These results show that the 1σ position and velocity errors (| mean | ± 1σ) are less than 10 m and 0.01 m/s, respectively. These results provide confidence that GOODS will meet Orbital's spacecraft real-time positioning requirement. It is recommended however that GOODS positioning specification remain at 20 m 1σ to account for real-world perturbations that may not have been simulated (e.g., ionospheric scintillation). Further ground evaluation will include closed-loop simulation using the Global Simulation Systems STR-4760 GPS simulator. Final validation of GOODS will occur on orbit in support of Orbview-3, to launch later this year. In this configuration, GOODS shall process live GPS measurements obtained via the onboard GPS receiver.

REFERENCES

1. MicroCosm Systems Description. Volume 1, Van Martin Systems, Inc., November 1998.

2. SA Model, "Global Positioning System: Theory and Applications, Volume I," B. W. Parkinson and P. Axelrod, ed., Chapter 17, 1996.

PERFORMANCE SIMULATION OF AUTONOMOUS SOLAR NAVIGATION*

Yanping Guo and Thomas E. Strikwerda
The Johns Hopkins University Applied Physics Laboratory

ABSTRACT

The performance of a new type of autonomous solar navigation system is analyzed in this paper. Such efficient autonomous navigation systems will reduce operation costs and alleviate the Deep Space Network workload in future space missions. The method is demonstrated by applying it to the STEREO mission. Orbit determination is simulated through the use of the mission-defined trajectory profile and solar angular data acquired by the on-board science instruments currently being considered. The study shows that the orbit solution derived by this new type of solar navigation system can satisfy the mission's navigation requirements; the position uncertainties obtained in the simulations are well below the mission allowable values, and are comparable to the results obtained with the conventional Doppler tracking system in some cases.

INTRODUCTION

Over the past decade, NASA's space missions have been evolved toward the direction of faster, better, and cheaper. More small missions have been launched recently, which are in the $50-100 million budget range with focused objectives and fast turn-around results, in contrast to the past when there were fewer but large missions with much higher budgets and long mission time. As the number of on-going missions increases, the Deep Space Network (DSN) will be overwhelmed with the DSN usage requests. A significant amount of time is needed for the Doppler tracking of the spacecraft to determine the orbit by the navigation team for each mission. It is therefore very desirable to develop more efficient navigation systems that have less or no dependence on the DSN; they can also minimize the requirements of ground operations.

A self-contained autonomous navigation system suggested in Ref. 1 uses the Sun as the navigation reference body and determines the spacecraft orbit based on observations made of the Sun using on-board instruments. The orbit is determined by tracking the directional change of the Sun by the spacecraft as it orbits the Sun or by measuring the optical Doppler shifts due to the relative motion of the spacecraft to the Sun. It has been demonstrated in Ref. 1 that the spacecraft orbit can be completely determined by either the directional data or the Doppler data alone, though the two types of data together may do a better job. This solar navigation system is best suited for missions with heliocentric orbits and for deep space missions during the long interplanetary cruise phase.

The performance of this new type of solar navigation system is analyzed in this paper by applying it to a specific space mission - the Solar TErrestrial RElations Observatory (STEREO) mission. Orbit determination is simulated using this mission as the study case. Two features of the STEREO mission are especially appealing for this new solar navigation system: the readily available on-board science instrument - the Solar Coronal Imaging Package (SCIP) - which can provide the Sun direction data for navigation; and the spacecraft's heliocentric orbit with no orbit maneuvers for the entire mission. In this study, it is assumed that the directional data only will be acquired for the spacecraft orbit determination.

MISSION PROFILE

STEREO is one of the Solar-Terrestrial Probe missions, a joint effort of NASA Goddard Space Flight Center and the Johns Hopkins University Applied Physics Laboratory. As described in the Report of the NASA Science Definition Team for the STEREO Mission (ref. 2), this mission will provide a new perspective view of the solar coronal mass ejection process and the heliospheric environment with 3D images, taken simultaneously

* This work is carried out under NASA-APL Prime Contract NAS5-97271, Task Order 14.

from two identical spacecraft traveling in near Earth orbits. While orbiting the Sun at a distance of 1 AU, one spacecraft will lead Earth and the other one will lag behind. After launched, the two spacecraft will gradually drift away from the Earth, forming favorable geometry for solar observations. The leading spacecraft will dwell near 20° (the Spacecraft-Sun-Earth angle) between 200 and 400 days into the mission, and near 45° between 600 and 800 days. The lagging spacecraft will dwell near 30° and 60°, respectively. No trajectory maneuvers are required, and the separation drifting is achieved by inserting the spacecraft into a heliocentric orbit each with a slightly different eccentricity from the Earth's orbit.

Each spacecraft will point its instrument boresight towards the Sun and orient the gimbaled high gain antenna towards the Earth within ±0.1 degrees for data communication. Each spacecraft carries a group of observation instruments including a Solar Coronal Imaging Package, which also provides the fine solar reference to the attitude control system of the spacecraft for maintaining the Sun pointing for solar observation. The mission requires that the instrument line of sight viewing the Sun should be maintained with a ±30 arc-seconds of 3σ accuracy. The spacecraft orbit position should be known better than 7500 km for science data analysis and ground station antenna pointing.

ACQUISITION OF THE SUN DIRECTION DATA

A coronagraph is a specialized solar observation instrument developed to view the solar corona by producing an artificial solar eclipse. It is essentially a telescope with an occulting disk in the focal plane to eclipse the image of the solar disk, and with other features to reduce stray sunlight so that the corona surrounding the occulting disk can be observed. Since the first flight on a sounding rocket in 1963 (ref. 3), the coronagraph has been frequently carried on spacecraft for various solar observation missions. Their spatial resolution, time resolution, mission duration, and observation range have been greatly improved over time.

Besides the primary function of producing the solar corona image, a coronagraph is also a fine sun sensor. High accuracy sun pointing data can be derived from the coronagraph measurements. It basically projects the sun disk image onto two pairs of detectors which measure the signal of the solar limb at four orthogonal positions, as shown in Fig. 1 (ref. 4, 5). Differentiating the signals of each pair of the detectors in line gives the center of the sun, providing the sun pointing direction relative to the spacecraft body fixed coordinate system. With the help of an on-board star tracker, the sun pointing direction may easily be referred to an inertial coordinate system. This sun pointing information is then fed to the guidance and control system of the spacecraft for maintaining an appropriate instrument pointing direction for desired science observations. Solar pointing direction with arc second accuracy can be easily achieved with current coronagraphs (ref. 5).

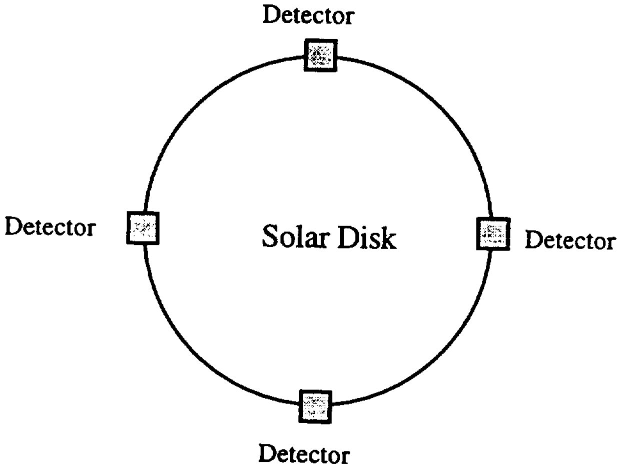


Figure 1 Measurement of Sun Direction with Four Orthogonal Detectors at the Limb of the Solar Image

For the STEREO mission, the on-board coronagraph instrument (SCIP) will provide the solar pointing knowledge accurate to 0.1 arc-sec (3σ) to the spacecraft's Attitude Control System (ref. 2), and the star tracker oriented anti-parallel to the instrument line of sight direction will have an accuracy of about 3 arc-sec (ref. 6). The simulations of navigation performance in this study assume the use of the on-board SCIP as the Sun sensor for acquiring the Sun direction tracking data and generate the simulated observation data with mission-defined instrument accuracy.

ORBIT DETERMINATION SIMULATION

Trajectory Generation

As described in the mission profile section, the STEREO mission will fly two spacecraft simultaneously. Since the trajectories of the two spacecraft are similar and navigation requirements are identical, only one spacecraft's orbit determination needs to be simulated. The results should be equally applicable to both. The leading spacecraft's performance is analyzed. The simulated trajectory for STEREO, in the ecliptic plane, together with the Earth's orbit, is shown in Fig. 2. The angular separation, i.e., the Spacecraft-Sun-Earth angle, and the range of the spacecraft from the Earth up to 800 days into the mission are shown in Fig. 3, and Fig. 4, respectively.

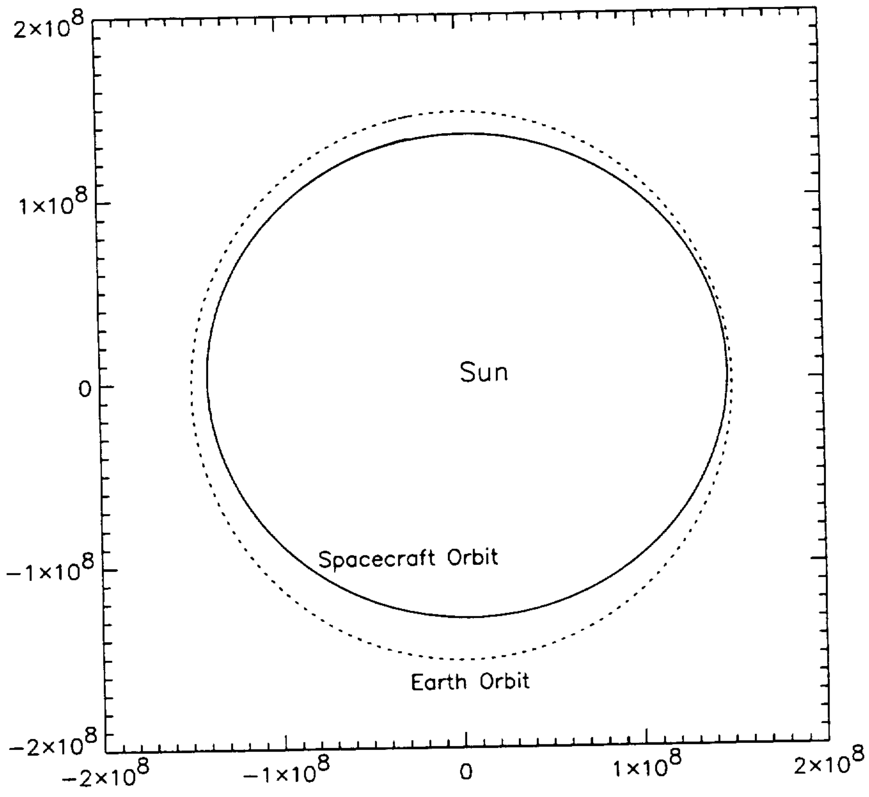


Figure 2 Spacecraft Trajectory in the Ecliptic Plane

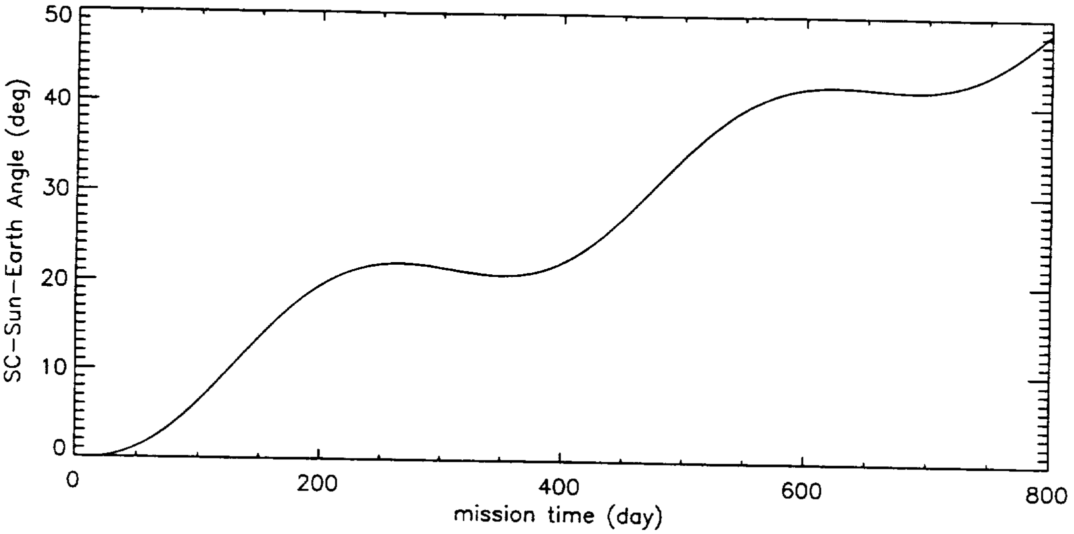


Figure 3 Angular Separation of the Spacecraft from the Earth

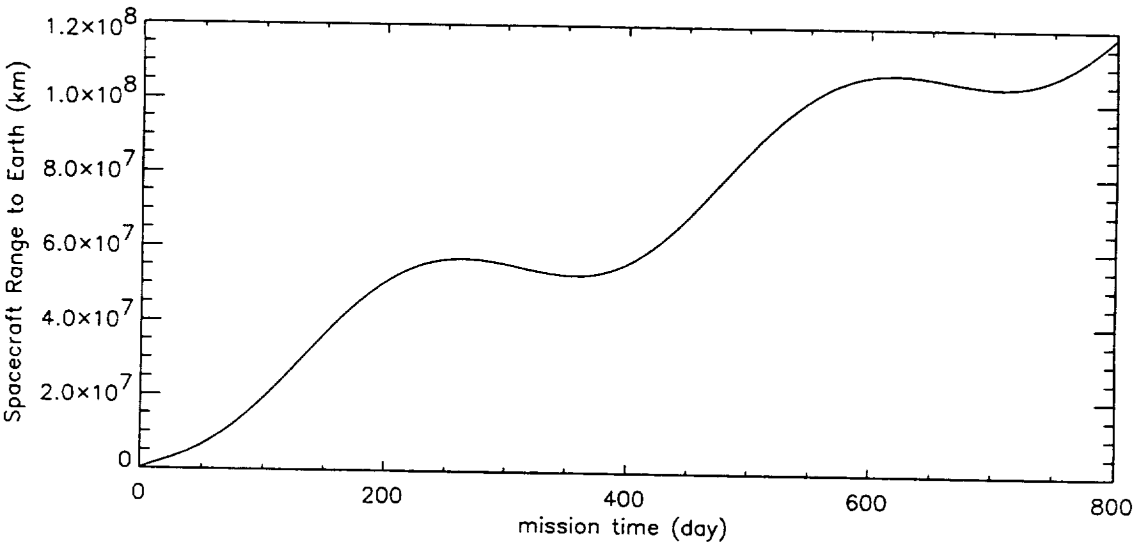


Figure 4 Spacecraft Distance from the Earth

Observation Data Generation

The Sun direction tracking data are generated based on the simulated spacecraft trajectory and the capability of the on-board instruments defined for the mission. Gaussian distributed white-noise errors are introduced to the assumed measurement data. A good star tracker can yield 10 μrad single frame accuracy (we have ignored for this study other errors such as uncorrected optical distortion and thermal effects). The direction data errors are dominated by the star tracker errors since the coronagraph errors are much smaller (0.16 μrad). A 10-μrad (1σ) error is used in generating the simulated observation Sun direction data.

Orbit Determination Program

An Orbit Determination (OD) program has been developed for estimating and predicting the spacecraft's position and velocity from the observed Sun direction data, which uses the least squares method minimizing the residuals between the observed quantities and the predicted ones. As demonstrated in Ref. 1 through theoretical analysis, a general spacecraft orbit specified by six classical orbit elements is completely determinable with the Sun direction data alone. For the simulation here, the orbit determination is performed through numerical computation by fitting the orbit parameters with observation data. The fitted parameters are the spacecraft's initial state vector, position and velocity. Observation data fed to the OD program are a series of time tagged Sun direction unit vectors.

Fitting Results and Features

Influence of Observation Coverage

Each observation data point contains a unit Sun direction vector at the observation time. It determines two of the six components of the spacecraft's state vector. Three such different data points will suffice to precisely determine the spacecraft's state if the observation data are error free. However, measurement errors are inevitable due to either the instruments or processing the measurements. Therefore, more data points are usually used in practical orbit determination to reduce the effect of measurement errors. In addition, some orbit parameters are insensitive to sun direction changes. Therefore, longer data arcs are necessary to accurately estimate them all.

As expected, the data coverage window does influence the estimation result substantially. Simulations show short data arc yields large uncertainty in the radial direction (the observation direction of the solar angular data) compared to the other two components. The angular data instantly fix the position vector in the plane perpendicular to the observation direction, but constrain the position vector along the observation direction through the accumulated angular changes over time. If we define the position error vector in the HLC coordinate system, a moving system with the unit vector H along the position vector direction, the unit vector C normal to the orbit plane, i.e., along the orbit angular momentum direction, and the unit vector L perpendicular to both H and C forming a right-handed system, the uncertainty ellipsoid of the position vector can be schematically represented as in Fig. 5.

The influence of the data arc length to orbit estimation is illustrated with three examples of the simulation, as shown in Fig. 6, where the residuals between the estimated spacecraft position and the true position are plotted in the HLC coordinate system. The estimated position is obtained by integrating the orbit using the estimated initial state vector derived from the OD program fed with the observation data. The three examples include: case A which is fitted with 10 days of data, case B with 20 days, and case C with 40 days. A data point is taken at every hour in the defined time span for all of the three cases. It shows that the along (L) and across (C) track components of the residuals remain small in all cases, but the radial (H) component errors reduced substantially when the data span is increased from 10 days to 40 days. The radial residuals reduced from 80 times larger (16000 km) to a level comparable to the other two components (200 km). It should be pointed out that with an orbit period of about 345 days, the data span of 10 day, 20 day, or 40 day, only covers a small fraction of the orbit circle, i.e., 1/34, 1/17 and 1/9, respectively. The simulation results are very encouraging in that an uncertainty of about 200 km in all three components are achieved with an observation window of only 1/9 of the orbit period, considering the measurement error of 10 μrad at the distance of 1 AU (1.5×10^8 km). This position uncertainty is much smaller than the mission allowable value of 7500 km.

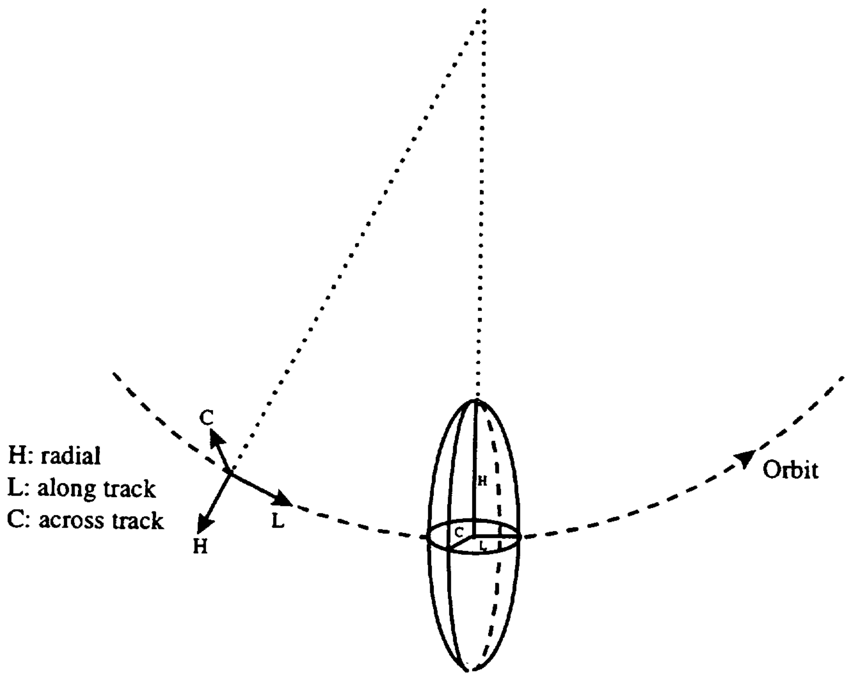


Figure 5 Position Error Ellipsoid in the HLC Coordinate System

Influence of Data Sampling Frequency

Effects of the data sampling frequency on orbit estimation are also examined by comparing the solutions in several cases in which observation data are provided with different data acquisition frequencies but having the same time span in length. Position residuals derived from observation data sampled in every 2 hours, every 4 hours, and every 12 hours in a time span of 40 days are plotted in Fig. 7. The trend is similar to the case of varying the time span length, in that the position errors are dominated by the H component and it increases significantly as the data density is reduced, while the other two components of the errors remain relatively steady, bounded by a few hundreds of kilometers. Although increased substantially in the H component, the position errors are still within the allowable value even if the data are sampled in a 40-day span with only two data points per day.

Convergence

Another important feature revealed in the simulations is that the fitting process performed by the OD program always converges very nicely. The final solution of the fitted parameters depends only on the observation data provided, its coverage range, sampling frequency, and error models, but is independent of the initially guessed values. The initial values guessed for the initial state vector input to the OD program appear to have no effects on the final solution. Good convergence means onboard autonomous orbit determination is probably robust. As an example, two different initial state vectors, as listed in Table 1, are input to the OD program fed with the same observation data file, and the exactly same solution of the estimated initial state vector is obtained.

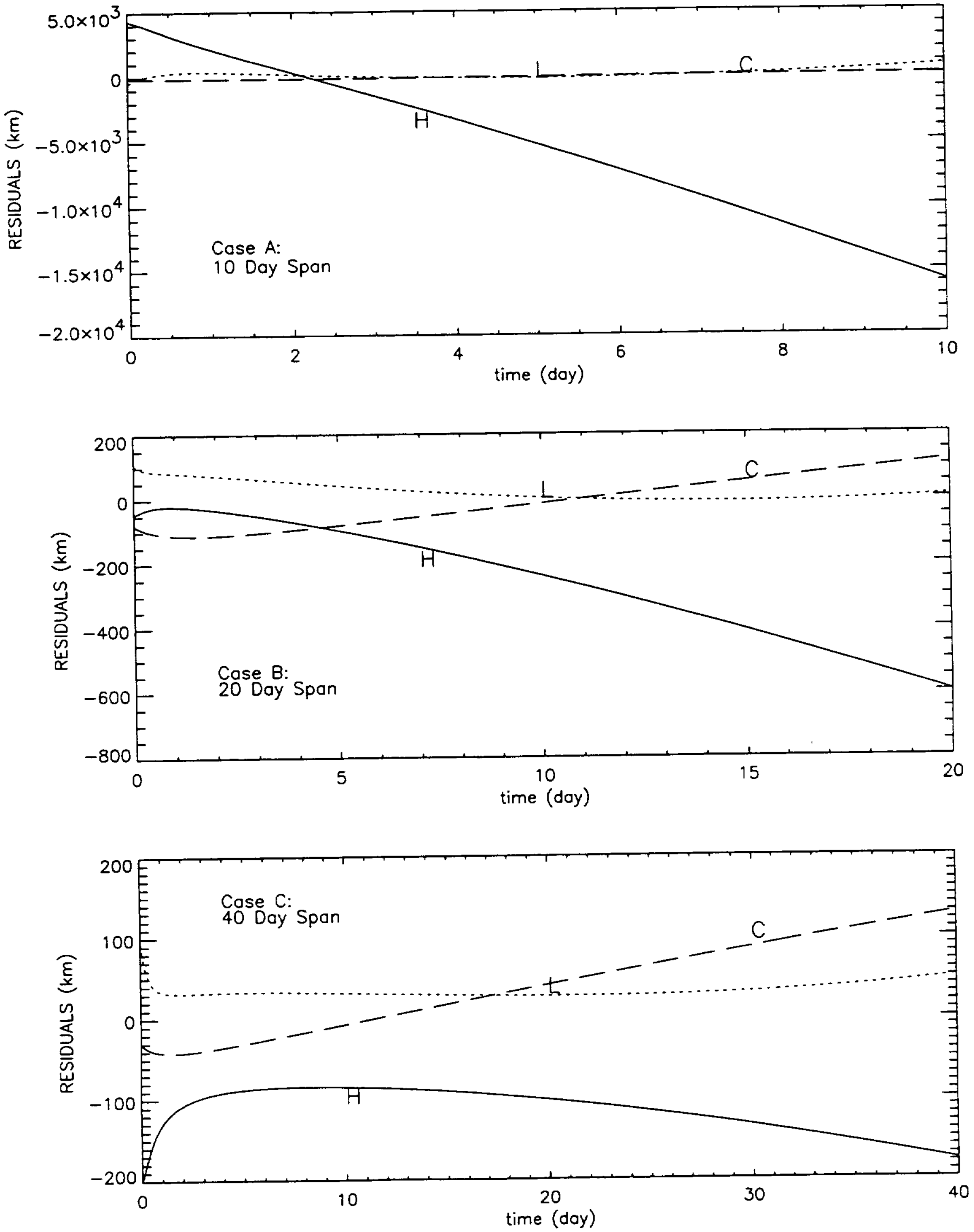


Figure 6 Effects of Data Span Length on Orbit Estimation: Residuals yielded in a 10-day span (top), 20-day span (center), and 40-day span (bottom)

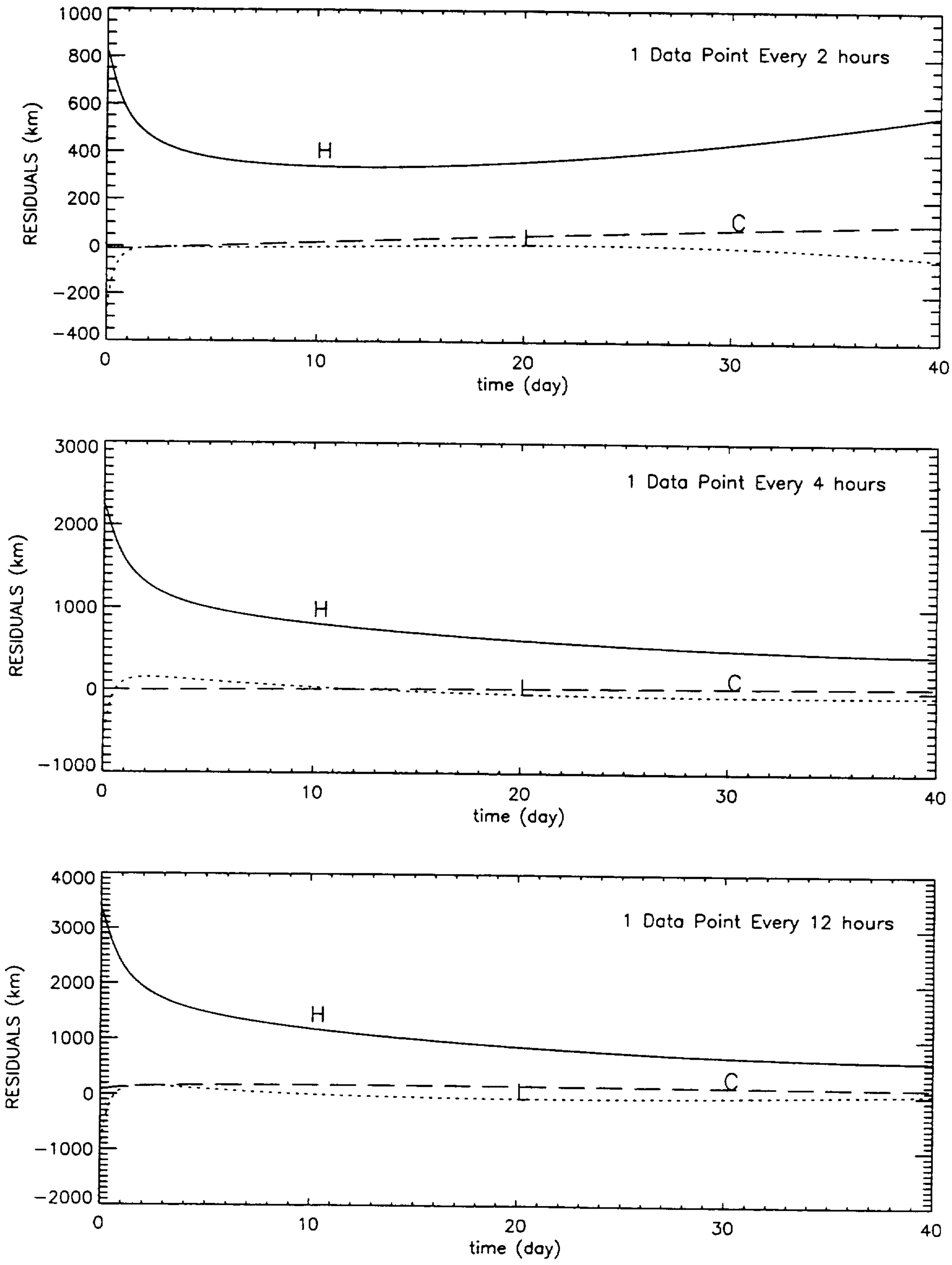


Figure 7 Effect of Data Sampling Frequency on Position Estimation: Residuals yielded in a 40-day span with data sampled in every 2 hours (top), every 4 hours (center), and every 12 hours (bottom)

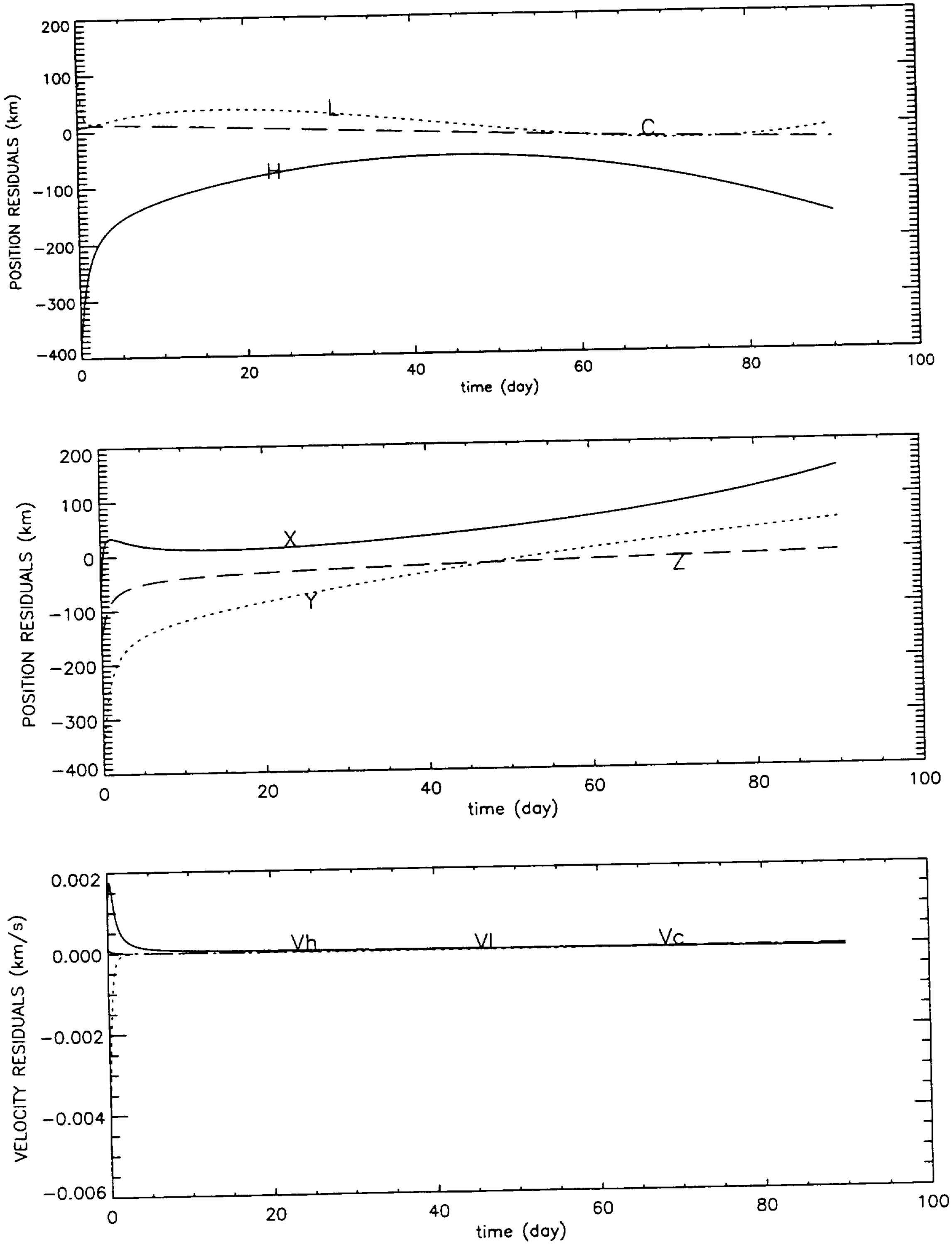


Figure 8 Estimation Results with 90-day Fitting Data: position residuals in HLC coordinates (top), position residuals in Cartesian coordinates (center), and velocity residuals in HLC coordinates (bottom)

Table 1. Different Initial Guesses Converge to the Same Estimated Initial State Vector

	Errors	X (km)	Y (km)	Z (km)	V_x (km/s)	V_y (km/s)	V_z (km/s)
Case 1	In Guessed State	5000.0	-5000.0	2000.0	0	0	0
	In Estimated State	-97.9378	-27.5948	-98.3366	0.001028	0.001268	-0.000404
Case 2	In Guessed State	50000.0	-50000.0	80000.0	-0.001	0.001	0.001
	In Estimated State	-97.9378	-27.5948	-98.3366	0.001028	0.001268	-0.000404

Residuals of orbit fitting with a longer data time span is shown in Fig. 8, where data arc of 90 days, about a quarter of the orbit circle, is used with data sampled at every hour. The results confirm the solution obtained with a shorter data span of 40 days, as shown in Fig. 6. This example further demonstrates the convergence stability of the new solar navigation system and its orbit determination program.

Orbit Solution

As indicated in both Fig. 6, orbit fitting results from a 40-day data span, and Fig. 8, results of a 90-day data span, an orbit solution better than about ±200 km is achievable with the solar navigation system using currently planned on-board instruments. The results are comparable to those baselined for STEREO with the conventional two-way Doppler tracking system: estimated uncertainties of ±200 km along track and ±100 km across track are cited in the mission report (ref. 2).

Further work will be carried out in the study including the effect of the solar pressure on spacecraft orbit, a more realistic noise model for the SCIP and star tracker, and the implementation of an on-board Kalman filter. The extension of this study will attempt to demonstrate that the inclusion of those additional error sources will still meet the 7500 km requirement.

CONCLUSION

The performance of a new type of self-contained autonomous solar navigation system, which determines the orbit based on observations of the Sun with on-board instruments, is studied by applying the system to the STEREO mission. Orbit determination studies are carried out using mission-defined trajectory and observation data simulated from the on-board science instrument and attitude subsystem. Simulation results reveal that orbit solutions comparable to the one derived by using the current ground-based two-way Doppler tracking system can be obtained with the new solar navigation system. With the obit determination program developed here, the orbit parameters fitting process can converge nicely, and its final solution appears to be independent of the initially guessed values for the initial state vector. It is found that an orbit solution better than about ±200 km is achievable with this new system using the on-board instruments. This is much better than the required uncertainty of ±7500 km. This analysis demonstrates that this new solar navigation system can satisfy the STEREO mission requirements, and is promising for achieving autonomous navigation in future space missions.

REFERENCES

1. Guo, Y., "Self-contained Autonomous Navigation System for Deep Space Missions," AAS paper, AAS 99-177, AAS/AIAA Space Flight Mechanics Meeting, Breckenridge, Colorado, February 7-10, 1999.

2. *The Sun and Heliosphere In Three Dimensions*, Report of the NASA Science Definition Team for the STEREO Mission, 1998.

3. Tousey, R., "Observations of the White-light Corona by Rocket," Ann. Astrophys. 28, 1965.

4. Scherrer, P. H., Bogart, R. S., Bush, R. I., Hoeksema, J. T., Kosovichev, A. G., Schou, J., Rosenberg, W., Springer, L., Tarbell, T. D., Title, A., Wolfson, C. J., Zayer, I., "The Solar Oscilations Investigation - Michelson Doppler Imager," Solar Physics, 162, 1995.

5. Rust, D. M., JHU/APL internal communication.

6. Ray, J. C. and Shapiro, H. S., *Guidance and Control*, STEREO Pre-Phase-A Requirements Review, 1999.

A Self-Tuning Kalman Filter for Autonomous Navigation Using the Global Positioning System (GPS)*

S. H. Truong
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland, USA 20771

Abstract

Most navigation systems currently operated by NASA are ground-based, and require extensive support to produce accurate results. Recently developed systems that use Kalman filter and GPS data for orbit determination greatly reduce dependency on ground support, and have potential to provide significant economies for NASA spacecraft navigation. These systems, however, still rely on manual tuning from analysts. A sophisticated neuro-fuzzy component fully integrated with the flight navigation system can perform the self-tuning capability for the Kalman filter and help the navigation system recover from estimation errors in real time.

1.0 Introduction

Autonomous navigation has the potential both to increase spacecraft navigation system performance and to reduce total mission cost. The Goddard Space Flight Center (GSFC) Flight Dynamics Analysis Branch (FDAB) has spent several years developing high-accuracy autonomous navigation systems for spacecraft using NASA's space and ground communications systems and enhanced these systems to support spacecraft using the Global Positioning System (GPS).

GSFC FDAB has developed navigation algorithms to meet a real-time accuracy goal of better than 20 meters (1σ) in position and 0.03 meter per second (1σ) in velocity using GPS Standard Positioning System (SPS) with selective availability (SA) corruption at typical levels. These algorithms, which are based on mature onboard navigation systems developed for spacecraft using NASA's space and ground communications systems, consist of the following core components:

- An extended Kalman filter (EKF) augmented with physically representative models for the gravity, atmospheric drag, solar radiation pressure, and time bias and drift state process noise to provide a realistic state error covariance
- A high-fidelity state dynamics model to reduce sensitivity to measurement errors and provide high-accuracy velocity estimates, permitting accurate state prediction during signal outages or degraded coverage
- Initialization and enhanced fault detection capabilities using instantaneous geometric GPS solutions

Detailed mathematical specifications for FDAB autonomous navigation systems using GPS are defined in Reference 1. Algorithms selected to meet the GPS navigation performance goals are summarized in Reference 2.

The FDAB has implemented these algorithms in a prototype GPS navigation software called the GPS Enhanced Orbit Determination Experiment (GEODE), which executes within the resource constraints of currently available flight processors (e.g., <400 kilobytes memory and <0.5 million instructions per second). Processing of raw pseudorange measurements from existing GPS receivers on the EP/EUVE and TOPEX/POSEIDON (T/P) spacecraft indicates that these navigation algorithms can provide accuracy of 10 meters (1σ) in total position and 0.01 meter per second (1σ) in total velocity with SA at typical levels. Without SA active, experiments performed in a realistically simulated flight environment produced converged solutions with errors of 15 meters maximum and 4 meters rms in total position. Improvements to the baseline algorithms to achieve real-time onboard accuracy of better than 2 meters (1σ) are discussed in Reference 2.

* This work is supported by the National Aeronautics and Space Administration (NASA)/Goddard Space Flight Center (GSFC), Greenbelt, Maryland, under a Research and Study Fellowship Program.

The core requirement for on-board autonomous navigation is a method of state estimation that handles uncertainties robustly, is capable of identifying estimation problems, flexible enough to make decisions and adjustments to recover from these problems, and compact enough to run on flight software. The current method of using EKF for state estimation requires manual tuning by support personnel. The re-tuning process is more complicated when dealing with geosynchronous or high-eccentricity orbits.

This paper discusses an approach to produce a high accuracy onboard navigation system that can recover from estimation errors in real time. The self-tuning capability is achieved by a neuro-fuzzy component augmented to the Kalman filter.

2.0 Extended Kalman Filter for Spacecraft Navigation

The orbit state estimation algorithm for FDAB autonomous navigation systems consists of an EKF that uses physically connected noise covariance models to account for force model and measurement errors. Autonomous navigation systems developed by the European Space Agency (ESA) or the Japanese Space Agency (NASDA) also use EKF for state estimation (References 3 and 4). The state vector consists of at least the user spacecraft position and velocity vectors. For GEODE, additional components include the atmospheric drag coefficient correction, solar radiation pressure coefficient correction, the GPS receiver time bias correction, and the time bias drift correction. The state vector estimation processing is performed at regular intervals, e.g., every 30 seconds, to propagate the filter state vector and covariance to the measurement time, update the state and covariance based on the measurements, and ouput telemetry data.

The state covariance matrix, [P], represents the filter uncertainty in the estimated state vector. It is initialized or reinitialized using ground uplinked parameters.

For GEODE, the state covariance [P] and the process noise covariance [Q] are [10X10] matrices, while the measurement noise covariance R is a scalar. To avoid the use of square roots and to guarantee nonnegativity of computed matrices, [P] and [Q] are factored into a unit upper triangular matrix [U] and a diagonal matrix [D], using Bierman's factorization methods. These [U] and [D] matrices are time propagated and measurement updated in the Kalman filter process, instead of [P] and [Q].

Parameters for [Q] and [R] are uplinked to the onboard navigation system to start or re-start the estimation process, or whenever the filter re-tuning is needed. For GEODE, there are ten parameters for [Q] and one parameter for [R]. Generally, parameters related to small unmodeled noises or to small errors in modeled accelerations that are not very well defined, are the ones to be updated in the re-tuning process. Several navigation fault detection tests are performed on the updated state and covariance. The Filter Convergence Test is the major test. If the filter has not converged and if the RSS position sigma, RSS velocity sigma, and semimajor axis sigma are all below their respective ground commandable convergence tolerances, then filter re-tuning is required. The current tuning process is performed by ground support analysts. Updated tuning parameters are uplinked to the onboard system to reset the filter.

3.0 Neuro-Fuzzy Systems

Neural networks and their learning capabilities have enjoyed a strong popularity with the development of the perceptrons in the 1960s and especially, after more powerful learning algorithms were discovered in 1985. A neural network is considered as a computing system that is made up of a number of simple, highly interconnected processing elements. Neural networks are used in many applications, from robot control to financial forecasting. A drawback of neural networks is that for some applications they do not always work as expected, and for the user a neural network simply is a black box. The user cannot learn from it.

Fuzzy logic is based on the idea of fuzzy sets, i.e. sets without clearly defined boundaries that can be used to model linguistic terms. Fuzzy systems associate with the process of formulating the mapping from a given input to an output using fuzzy logic that provides a basis from which decisions can be made, or patterns discerned. Fuzzy systems can be used for the same tasks as neural networks. They are successfully applied in fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. Fuzzy systems are not, however, created by a learning algorithm. A major problem is that its parameters must be tuned manually, usually in an error-prone and time-consuming process.

Neuro-fuzzy systems are built from the idea of applying neural network algorithms to automatically determine and tune parameters of fuzzy systems. That combination could avoid drawbacks of both neural networks and fuzzy systems, and constitutes an interpretable model that is capable of learning and using problem-specific prior knowledge.

Various neuro-fuzzy models have been developed. The Adaptive Neuro Fuzzy Inference System (ANFIS) model and its generalization for multiple inputs/outputs systems are used to prototype the self-tuning component for autonomous navigation using Kalman filter. This preliminary choice is mainly based on the model efficiency, software availability, and the fuzzyness of filter outputs. The final product is expected to be more complex.

Several neuro-fuzzy system models are described with details in References 5 and 6. Fig. 1 shows the architecture of a two-input ANFIS with nine rules.

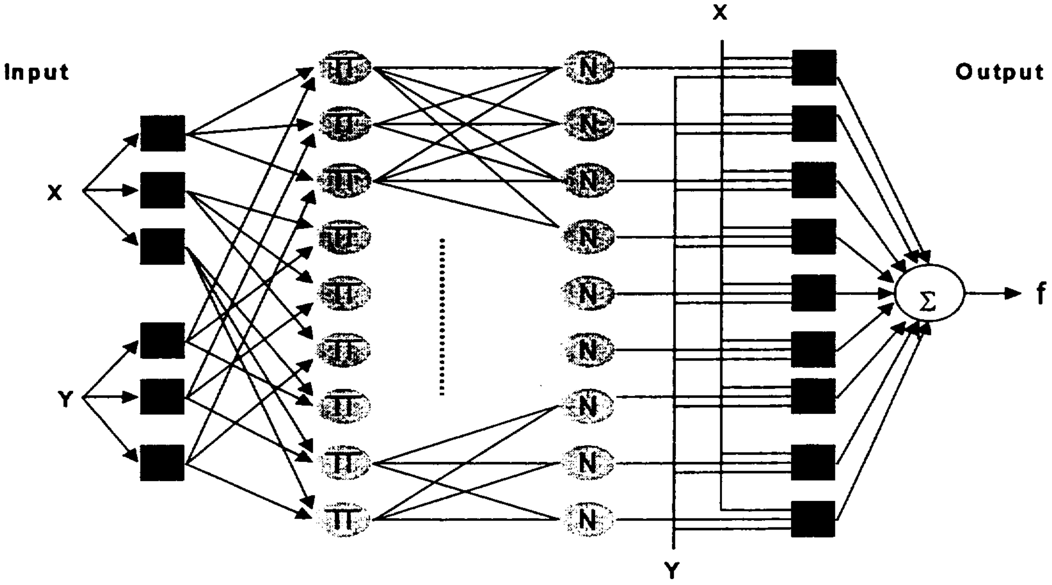


Figure 1. Architecture of a Two-Input ANFIS with Nine Rules

4.0 Neuro-Fuzzy System for a Self-Tuning EKF

The self-tuning method discussed in this paper is to optimize navigation autonomy for GEODE, which uses GPS as a main tracking system. This method, however, can be applied without significant modifications to any other system that uses a Kalman filter for autonomous navigation.

Fig. 2 illustrates a high-level architecture of the integrated system.

Outputs from the filter include the state error covariance matrix [P], measurement residual [M], and appropriate information relating to the filter and satellite status. [P] and [M] are gathered in time series, limited by a reasonable preset time window. When the filter is not convergent and covariances hit preset thresholds, which are less generous than those set by the Filter Convergence Test, the re-tuning process is needed. Functional representations for the [P], [M] time series are then determined (e.g., using a least-squares polynomial fitting), and the preprocessor forms an input vector to the neuro-fuzzy system. The neuro-fuzzy system analyzes these inputs to produce tuning data to be used to adjust [Q] and [R].

Input patterns and target parameters are specifically modeled to train the neuro-fuzzy system for a given user spacecraft. The training process is performed prior to the operational use of the system.

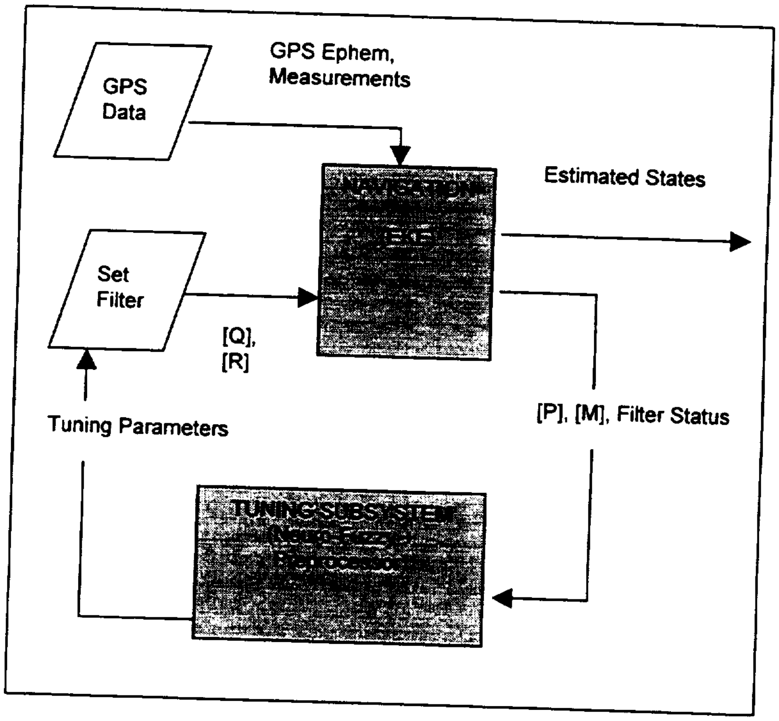


Figure 2. High-Level Architecture of the Self-Tuning Kalman Filter for Autonomous Navigation Using GPS

5.0 Prototype for Phase I Development

The real scenario of the self-tuning navigation system can be much more complex than as described above and the final product would be capable to respond dynamically to filter problems. The main issue, however, is simply to find a mapping between the behavior of the filter output (e.g., state error covariance) and the tuning parameters. The primary phase of the development of the self-tuning Kalman Filter for autonomous navigation is therefore to build a simple prototype that can prove the existence of such a mapping. The target navigation system for this prototype is GEODE processing GPS data for low earth orbits (LEO). For LEO user spacecraft, there are three parameters that are related to errors in the acceleration models for solar gravity and lunar gravity; or to unmodeled accelerations from polar motion, tidal effects, random venting, etc. These parameters need to be updated via the tuning process. Preliminary examination of output data from different cases shows that patterns of velocity variances (or standard deviations) are adequate in the determination of tuning parameters. The tuning subsystem prototype for Phase I is simply a three inputs/three outputs neuro-fuzzy system augmented by a preprocessor that gathers filter outputs (i.e. state error covariance) in time series, determines if the filter re-tuning is needed, and uses least-squares process to fit them to second degree polynomials. The preprocessor also builds a vector that functionally represents the behavior of the covariance and that is input to the neuro-fuzzy system. Parameters are tuned using the hybrid option that is a mixture of least-squares and backpropagation techniques. An asymmetric and closed sigmoidal function is used as a membership function.

Fig. 3 shows a high-level diagram of the Phase I prototype.

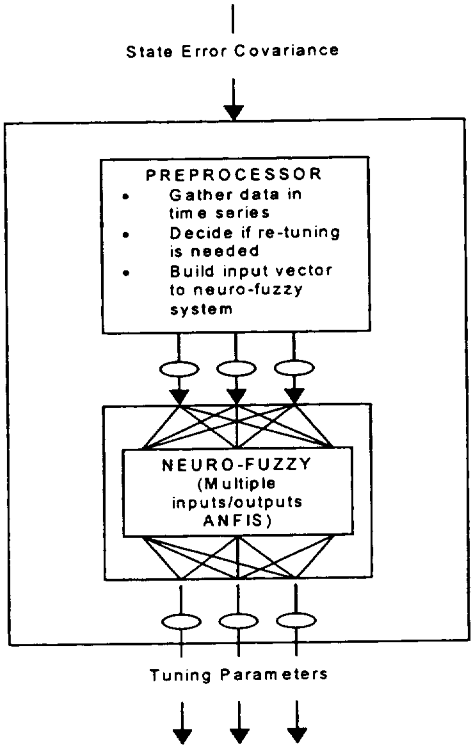

State Error Covariance
PREPROCESSOR
• Gather data in time series
• Decide if re-tuning is needed
• Build input vector to neuro-fuzzy system
NEURO-FUZZY (Multiple inputs/outputs ANFIS)
Tuning Parameters

Figure 3. High-Level Diagram of the Phase I Prototype

6.0 Test Results

Data from the GEODE processing of real GPS pseudorange measurement with SA on, obtained from an experimental receiver flown on the TOPEX/POSEIDON (T/P) spacecraft on November 17, 1993, were used to test the Phase I prototype.

Fig. 4 shows the convergence of the in-track velocity standard deviation from the T/P testing. Similar curves are seen in other components as well as in the corresponding position standard deviations. This behavior reflects a filter status where correct tuning parameters are provided.

To train the neuro-fuzzy system, standard deviation patterns and corresponding target tuning parameters for fifty cases are used. Fig. 5 shows 3 patterns that correspond to different errors in one of three tuning parameters, Qi.

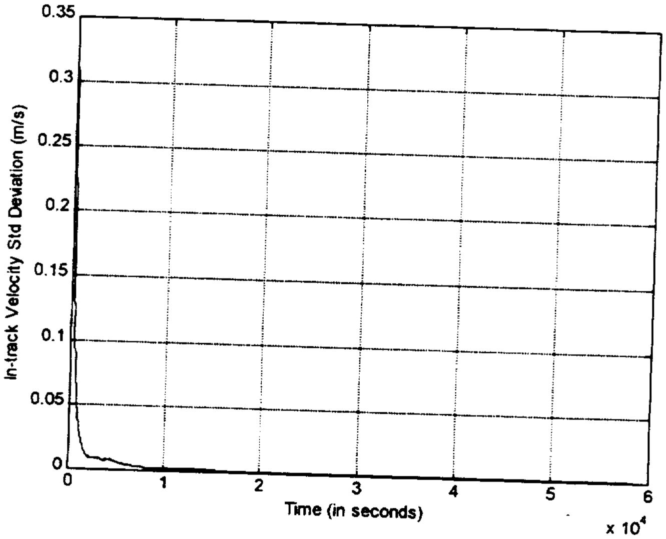


Figure 4. In-track Velocity Standard Deviation from the T/P Testing

Results from preliminary testing of this Phase I prototype show that errors in tuning parameters are identified and the system can be well recovered from these errors. The average testing error is 0.0024 m/s for parameters ranging from 0.02 to 0.8 m^2/s^3. Fig. 6 shows the average difference between the in-track velocity standard deviations obtained from the correct Qi and from that determined by the prototype.

These test results are encouraging for this preliminary work. It is still premature, however, to have a good conclusion about the quality and practicality of this method of self-tuning when applying to the complex operational scenario of a real autonomous navigation system.

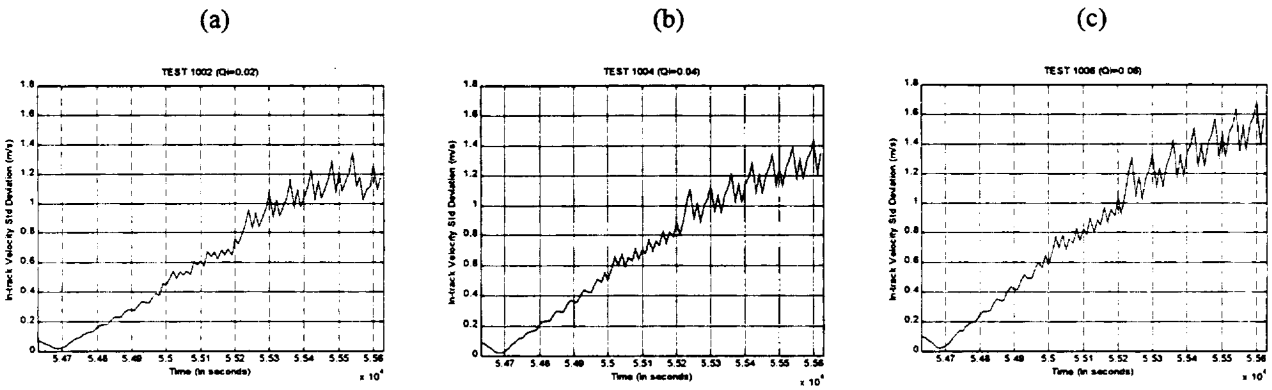


Figure 5. In-track Velocity Standard Deviations from T/P Testing for Different Errors in a Tuning Parameter

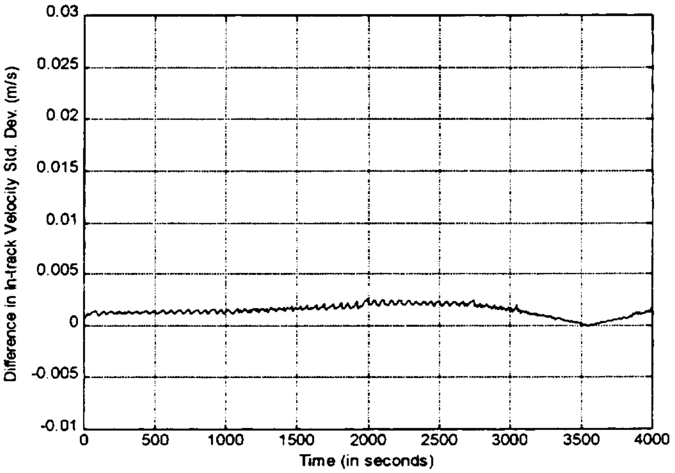


Figure 6. Average Error in Phase I Prototype Testing (for In-track Velocity Standard Deviation)

6.0 Future Directions

Phase II of the development of the self-tuning Kalman Filter for autonomous navigation is to refine the self-tuning method to accommodate to a much more complex operational scenario and to accordingly complete the system prototype.

Phase III will involve the extension of the self-tuning filter to cover geosynchronous spacecraft and high- eccentricity orbits. For these cases, more parameters need to be updated in the re-tuning process and the tuning frequency is projected to be much higher.

Acknowledgment

The author wishes to acknowledge Taesul Lee and Ben Lin of Computer Sciences Corporation (CSC) for their assistance in the testing of GEODE.

References

1. Goddard Space Flight Center, Flight Dynamics Division, 553-FDD-95/013R1UD0, *Global Positioning System (GPS) Enhanced Orbit Determination Experiment (GEODE) Mathematical Specifications*, T. Lee and A. C. Long, March 1997
2. Son H. Truong et al., "High Accuracy Autonomous Navigation Using the Global Positioning System (GPS)," *Proceedings of the 12th International Symposium on Space Flight Dynamics*, Darmstadt, Germany, 2-6 June 1997, pp. 73-78
3. J.-P. Berthias et al., "General Characteristics of Real-time On-board Orbit Determination," *Proceedings of the 12th International Symposium on Space Flight Dynamics, Darmstadt*, Germany, 2-6 June 1997
4. Maki Maeda et al., "NASDA's Precise Orbit Determination System," *Proceedings of the 14th International Symposium on Space Flight Dynamics, Iguassu Falls*, Brazil, 8-12 February 1999
5. Detlef Nauck et al., *Foundations of Neuro-Fuzzy Systems*. England: John Wiley & Sons, 1997
6. J.-S. R. Jang et al., *Neuro-Fuzzy and Soft Computing*. New Jersey: Prentice Hall, 1997

AN ALTERNATIVE LUNAR EPHEMERIS MODEL FOR ON-BOARD FLIGHT SOFTWARE USE

David G. Simpson
Flight Software Branch
NASA Goddard Space Flight Center, Greenbelt, Maryland 20771

ABSTRACT

In calculating the position vector of the Moon in on-board flight software, one often begins by using a series expansion to calculate the ecliptic latitude and longitude of the Moon, referred to the mean ecliptic and equinox of date. One then performs a reduction for precession, followed by a rotation of the position vector from the ecliptic plane to the equator, and a transformation from spherical to Cartesian coordinates before finally arriving at the desired result: equatorial J2000 Cartesian components of the lunar position vector. An alternative method is developed here in which the equatorial J2000 Cartesian components of the lunar position vector are calculated directly by a series expansion, saving valuable on-board computer resources.

INTRODUCTION

The calculation of the orbit of the Moon is one of the oldest problems in celestial mechanics. Its solution has had great historical significance as a test of Newton's theory of gravity, with much of the early work on the problem having been done by Newton himself in his discussion of the two- and three-body problems in Book I of the *Principia*. In past centuries, accurate predictions of the position of the Moon have also been of great practical interest as a navigational aid for seafaring vessels, prompting the English government and scientific societies to offer rewards for accurate lunar prediction tables.[1] The resulting body of work developed during the eighteenth and nineteenth centuries forms the basis of the lunar theory still in use today.

Modern lunar theory was first developed by G.W. Hill[2–5] in 1878, and later expanded and improved by E.W. Brown[6] in 1896. The problem of lunar motion addressed by Hill and Brown is a surprisingly difficult one; while the underlying physical laws are very simple, the motion itself is quite complex.[7–11] The basic motion of the Moon around Earth is affected by many strong perturbations such as those due to the Sun, the other planets, and Earth's equatorial bulge. These perturbations result in an advancement of the line of apsides of the lunar orbit, a regression of the line of nodes, and other periodic perturbations superimposed on these motions. For high accuracy, it is necessary to compute hundreds of periodic variations in the motion, although computing only the most important few terms results in a level of accuracy that is adequate for flight software use.

There have been two major reasons for calculating the position of the Moon in spacecraft on-board computer flight software. First, one often wishes to write flight software to prevent the spacecraft from pointing sensitive instruments at the Moon, which can have an apparent magnitude as bright as −12 at full Moon.[12] Second, one may require the flight software to calculate stellar aberration corrections.[13] For high accuracy, this requires calculating the velocity vector of Earth with respect to the Earth-Moon barycenter, which in turn requires a calculation of the lunar velocity vector. If the flight software can calculate a lunar position vector, then this velocity vector may be found by differentiating the lunar position vector with respect to time.

REVIEW OF CURRENT MODELS

A number of approaches for calculating a lunar position vector are currently used by spacecraft flight software. In the flight software for the Hubble Space Telescope's DF-224 flight computer, for example, one finds the position of the Moon using a simple two-body model. The standard two-body calculations[14] are modified somewhat to allow for the motion of the nodes and apsides of the lunar orbit. A new set of orbital elements is uplinked from the ground every few days to keep the error in the model to within acceptable limits, on the order of 1°. While this model is not highly accurate, it has the virtue of being very fast—a necessity for the 1970s-vintage flight computer.

An approach commonly used with more modern flight computers is based on the low-precision formulae given in the *Astronomical Almanac*.[15,16] This model is based on earlier work done by the Almanac Offices of the United States and United Kingdom[17] and by Eckert, Walker, and Eckert,[18] all of which are based on Brown's lunar theory.[6] In this model, one begins by using series expansions to calculate the ecliptic longitude λ, ecliptic latitude β, and horizontal parallax π of the Moon, referred to the mean ecliptic and equinox of date:

$$\begin{aligned}\lambda \;=\;& 218\overset{\circ}{.}32 + 481\,267\overset{\circ}{.}883\ t\\ &+6\overset{\circ}{.}29\sin(477\,198\overset{\circ}{.}85\ t + 134\overset{\circ}{.}9)\\ &-1\overset{\circ}{.}27\sin(-413\,335\overset{\circ}{.}38\ t + 259\overset{\circ}{.}2)\\ &+0\overset{\circ}{.}66\sin(890\,534\overset{\circ}{.}23\ t + 235\overset{\circ}{.}7)\\ &+0\overset{\circ}{.}21\sin(954\,397\overset{\circ}{.}70\ t + 269\overset{\circ}{.}9)\\ &-0\overset{\circ}{.}19\sin(35\,999\overset{\circ}{.}05\ t + 357\overset{\circ}{.}5)\\ &-0\overset{\circ}{.}11\sin(966\,404\overset{\circ}{.}05\ t + 186\overset{\circ}{.}6)\ ,\end{aligned} \tag{1}$$

$$\begin{aligned}\beta \;=\;& +5\overset{\circ}{.}13\sin(483\,202\overset{\circ}{.}03\ t + 93\overset{\circ}{.}3)\\ &+0\overset{\circ}{.}28\sin(960\,400\overset{\circ}{.}87\ t + 228\overset{\circ}{.}2)\\ &-0\overset{\circ}{.}28\sin(6\,003\overset{\circ}{.}18\ t + 318\overset{\circ}{.}3)\\ &-0\overset{\circ}{.}17\sin(-407\,332\overset{\circ}{.}20\ t + 217\overset{\circ}{.}6)\ ,\end{aligned} \tag{2}$$

$$
\begin{aligned}
\pi \;=\; & 0^\circ\!.9508 \\
& +0^\circ\!.0518\cos(477\,198^\circ\!.85\; t + 134^\circ\!.9) \\
& +0^\circ\!.0095\cos(-413\,335^\circ\!.38\; t + 259^\circ\!.2) \\
& +0^\circ\!.0078\cos(890\,534^\circ\!.23\; t + 235^\circ\!.7) \\
& +0^\circ\!.0028\cos(954\,397^\circ\!.70\; t + 269^\circ\!.9)\ .
\end{aligned}
\tag{3}
$$

The horizontal parallax π gives the Earth-Moon distance r:

$$
r \;=\; \frac{R_\oplus}{\sin\,\pi}\ , \tag{4}
$$

where $R_\oplus = 6378.140$ km is the equatorial radius of Earth (IAU 1976 value).[19]

Having found the lunar ecliptic mean-of-date coordinates, one must then perform a reduction for precession to epoch J2000 (2000 January 01 12:00:00 Barycentric Dynamical Time) to find the ecliptic J2000 coordinates $(\lambda_0,\ \beta_0)$. To sufficient precision, this may be found using the formulae[20]

$$
\beta_0 \;=\; \beta - b\sin(\lambda + c)\ , \tag{5}
$$

$$
\lambda_0 \;=\; \lambda - a + b\cos(\lambda + c)\tan\beta_0\ , \tag{6}
$$

where the precession constants a, b, and c are given by

$$
a \;=\; 1^\circ\!.396\,971\; t + 0^\circ\!.000\,3086\; t^2\ , \tag{7}
$$

$$
b \;=\; 0^\circ\!.013\,056\; t - 0^\circ\!.000\,0092\; t^2\ , \tag{8}
$$

$$
c \;=\; 5^\circ\!.123\,62 - 1^\circ\!.155\,358\; t - 0^\circ\!.000\,1964\; t^2\ , \tag{9}
$$

and where t is the time in Julian centuries (cy) of 36 525 days from J2000:

$$
t = (\mathrm{JDE} - 245\,1545.0)/36\,525\ , \tag{10}
$$

and JDE is the ephemeris Julian day.

The remaining step is to rotate the coordinates from the plane of the mean ecliptic of J2000 to the mean equator of J2000, and to convert from spherical polar to Cartesian coordinates:

$$
X \;=\; r\cos\beta_0\cos\lambda_0\ , \tag{11}
$$

$$
Y \;=\; r(\cos\beta_0\sin\lambda_0\cos\varepsilon_0 - \sin\beta_0\sin\varepsilon_0)\ , \tag{12}
$$

$$
Z \;=\; r(\cos\beta_0\sin\lambda_0\sin\varepsilon_0 + \sin\beta_0\cos\varepsilon_0)\ , \tag{13}
$$

where r is given by Eq. (4) and $\varepsilon_0 = 23^\circ\ 26'\ 21''\!.448$ is the obliquity of the ecliptic at J2000 (IAU 1976 value).[21]

This model has very good precision for on-board flight software use: the rms error in the lunar position is about $0^\circ\!.11$, with a maximum error of about $0^\circ\!.35$.

A NEW MODEL

Many of the equations involved in computing the position of the Moon using the method just described involve what is essentially a coordinate transformation, from ecliptic mean-of-date coordinates to equatorial J2000 Cartesian coordinates. In this paper, I investigate the possibility of calculating the equatorial J2000 Cartesian coordinates directly by series expansions similar to Eqs. (1–3), thus eliminating the need for performing the coordinate transformations in on-board flight software.

We begin by assuming that each of the J2000 equatorial Cartesian coordinates X_n may be represented by Fourier sine series:

$$X_n = \sum_{m=1}^{N_n} a_{nm} \sin(\omega_{nm}\, t + \delta_{nm}) \; , \tag{14}$$

where $X_1 \equiv X$, $X_2 \equiv Y$, and $X_3 \equiv Z$; N_n is the order of the series for X_n. We now need to find the amplitudes a_{nm}, frequencies ω_{nm}, and phase constants δ_{nm}. This may be done by fitting these parameters to the DE200 ephemeris model[22,23] using an exhaustive search. DE200 is an ephemeris model developed at the Jet Propulsion Laboratory, and has been used to produce tables in the *Astronomical Almanac* since 1984. It calculates Cartesian coordinates of Solar System objects, referred directly to the mean equator and equinox of J2000.

For each coordinate, the terms of the series in Eq. (14) may be found one at a time by simultaneously fitting the parameters a_{nm}, ω_{nm}, and δ_{nm} over a grid of possible values to the DE200 model. An algorithm for accomplishing this involves calculating the error $\epsilon_{a\omega\delta}$ between the DE200 model and a "test model" $a \sin(\omega t + \delta)$ using each combination of parameters a, ω, and δ:

```
for $a$ = $a_{min}$ to $a_{max}$
  for $\omega$ = $\omega_{min}$ to $\omega_{max}$
     for $\delta$ = $\delta_{min}$ to $\delta_{max}$
        $\epsilon_{a\omega\delta}$ = $\sum_{t=2000}^{2100}[X_{DE200}(t) - a\sin(\omega t + \delta)]^2$ ,
```

where the summation is over 2^{16} points covering the interval A.D. 2000–2100. The smallest error $\epsilon_{a\omega\delta}$ found gives the best fit parameters a, ω, and δ. This process may be repeated several times over successively smaller search ranges and finer grid spacings in order to find more significant digits for the parameters. Once a term has been found, it is subtracted from the DE200 data, and the whole process repeated on the remaining data to find the next term in the series.

In the model given by Eq. (14), we assume that the amplitudes a_{nm} are all positive, so that amplitudes may be searched over a grid of values between 0 and the maximum in the data set. The amplitudes may be assumed to be positive without loss of generality by allowing the phase constants δ_{nm} to be searched over the entire range 0 to 2π: since $-\sin\theta \equiv \sin(\theta + \pi)$, any potential minus sign in the amplitude is simply absorbed as an extra π radians added to the phase constant.

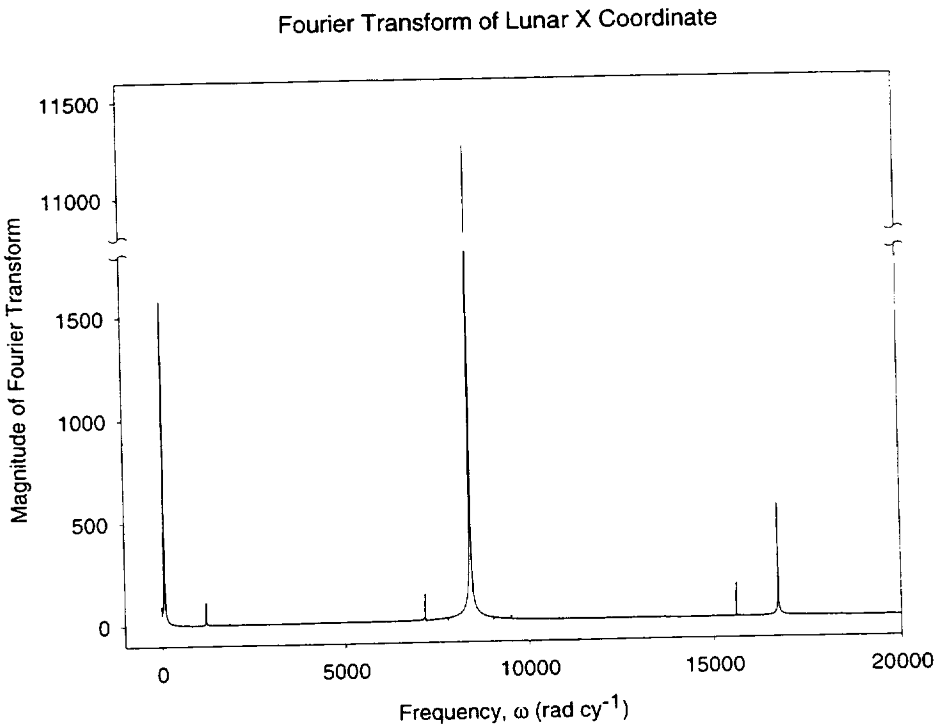


Figure 1. Fourier spectrum of lunar X coordinate (A.D. 2000–2100).

Determining a search range for the frequencies ω_{nm} is somewhat more complicated than it is for the amplitudes and phase constants. A search range for ω_{nm} may be determined by examining the peaks in the Fourier transform $\hat{X}_n(\omega)$ of the DE200 data:

$$\hat{X}_n(\omega) = \int_{-\infty}^{\infty} X_n(t)\, e^{i\omega t}\, dt \ , \tag{15}$$

where $X_n(t)$ is the position coordinate at time t, and ω is the angular frequency. This Fourier transform may be calculated by using the DE200 model to compute the lunar position vector at N discrete time points t_k, then finding the discrete Fourier transform $\hat{X}_n(\omega_p)$:

$$\hat{X}_n(\omega_p) = \sum_{k=0}^{N-1} X_n(t_k)\, e^{i\omega_p t_k} \ , \tag{16}$$

where $X_n(t_k)$ is the position vector at time point t_k, $\omega_p = 2\pi p/t_N$ is the angular frequency, and $p = 0, 1, 2, \ldots, N-1$. For this study, $N = 2^{14}$ time points were chosen over the time interval A.D. 2000–2100; the magnitude of the resulting Fourier transform $|\hat{X}_1(\omega_p)|$ for X is shown in Figure 1. For each term in the series expansion (Eq. 14), a search range is taken around one of the peaks in the Fourier spectrum.

This exhaustive search process, which is essentially a curve fit to the DE200 model, required about one week of computer time to find each term in a series, and some five months of computer time to find the complete solution to seven terms per series. The final results are:

$$\begin{aligned} X \;=\;& 383.0\ \sin\,(8399.685\,t\ +\ 5.381) \\ &+ 31.5\ \sin\,(70.990\,t\ +\ 6.169) \\ &+ 10.6\ \sin\,(16\,728.377\,t\ +\ 1.453) \\ &+ 6.2\ \sin\,(1185.622\,t\ +\ 0.481) \\ &+ 3.2\ \sin\,(7143.070\,t\ +\ 5.017) \\ &+ 2.3\ \sin\,(15\,613.745\,t\ +\ 0.857) \\ &+ 0.8\ \sin\,(8467.263\,t\ +\ 1.010)\ \times 10^6\ \mathrm{m}\,, \end{aligned} \tag{17}$$

$$\begin{aligned} Y \;=\;& 351.0\ \sin\,(8399.687\,t\ +\ 3.811) \\ &+ 28.9\ \sin\,(70.997\,t\ +\ 4.596) \\ &+ 13.7\ \sin\,(8433.466\,t\ +\ 4.766) \\ &+ 9.7\ \sin\,(16\,728.380\,t\ +\ 6.165) \\ &+ 5.7\ \sin\,(1185.667\,t\ +\ 5.164) \\ &+ 2.9\ \sin\,(7143.058\,t\ +\ 0.300) \\ &+ 2.1\ \sin\,(15\,613.755\,t\ +\ 5.565)\ \times 10^6\ \mathrm{m}\,, \end{aligned} \tag{18}$$

$$\begin{aligned} Z \;=\;& 153.2\ \sin\,(8399.672\,t\ +\ 3.807) \\ &+ 31.5\ \sin\,(8433.464\,t\ +\ 1.629) \\ &+ 12.5\ \sin\,(70.996\,t\ +\ 4.595) \\ &+ 4.2\ \sin\,(16\,728.364\,t\ +\ 6.162) \\ &+ 2.5\ \sin\,(1185.645\,t\ +\ 5.167) \\ &+ 3.0\ \sin\,(104.881\,t\ +\ 2.555) \\ &+ 1.8\ \sin\,(8399.116\,t\ +\ 6.248)\ \times 10^6\ \mathrm{m}\,, \end{aligned} \tag{19}$$

where all angles are given in *radians* for convenience of use in software, t is the time in Julian centuries from J2000 given by Eq. (10), and X, Y, and Z are the Cartesian components of the lunar position vector, referred to the mean equator and equinox of J2000. The terms are arranged in order of decreasing contribution to the reduction in the error of the model.

One of the primary advantages of this model is that it allows a lunar ephemeris to be programmed in flight software using very little code. Using Eqs. (17–19), an entire lunar ephemeris model may be programmed in just a few lines of C code:

```
for (n=0; n<3; n++)
   {
   x[n] = 0.0;
   for (m=0; m<7; m++)
      x[n] += a[n][m]*sin(w[n][m]*t+delta[n][m]);
   }
```

Calculations for the reduction for precession, rotation from the ecliptic to the equator, and transformation from spherical polar to Cartesian coordinates have essentially been "absorbed" into the series coefficients, and so do not need to be performed explicitly.

DISCUSSION OF THE NEW MODEL

An examination of the frequencies in the terms of the *Astronomical Almanac* model of Eqs. (1–3) and of the new model of Eqs. (17–19) gives some interesting insights into the lunar motion. The frequencies in the *Astronomical Almanac* model are all computed as functions of the mean anomalies and mean longitudes of the Sun and Moon,[16] while the frequencies in the model given by Eqs. (17–19) are determined entirely by a curve fit. We examine the origins of some of the more prominent frequencies in both models below.

Anomalistic Month

The dominant term in the expressions for the ecliptic longitude λ (Eq. 1) and horizontal parallax π (Eq. 3) have a frequency of 477 198.85 deg cy^{-1}. In deriving the *Astronomical Almanac* series, this frequency was computed as the rate of change of the Moon's mean anomaly. Since the mean anomaly is measured in the plane of the orbit from the perigee point, one complete cycle of the mean anomaly requires the same amount of time as the Moon's motion from its perigee point to its next perigee. It comes as no surprise, then, that this frequency of 477 198.85 deg cy^{-1} is equal to one revolution per *anomalistic month* of 27.554 550 days, where an anomalistic month is the time required for the Moon to move from perigee to perigee.

Draconic Month

For the ecliptic latitude β (Eq. 2), the dominant term has a frequency of 483 202.03 deg cy^{-1}. This was computed as the rate of change of the Moon's mean longitude, which is measured from the vernal equinox to the ascending node along the ecliptic plane, then from the node to the Moon along the orbit plane. The Moon will have $\beta = 0$ only when it is at one of the nodes of the orbit, and it will next have $\beta = 0$ again (crossing the node in the same direction) when it returns to the same node again. We might therefore expect that the dominant term in the expression for the ecliptic latitude will be the time required for the Moon to move from an orbital node back to the same node. Indeed, the frequency of 483 202.03 deg cy^{-1} is equal to one revolution per *draconic month* of 27.212 221 days, where a draconic month is the time required for the Moon to move from an orbital node back to the same node.

Sidereal Month

In the series for X, Y, and Z in the new model (Eqs. 17–19), on the other hand, the dominant terms all have a frequency of about 8399.685 rad cy^{-1}, which is equal to 1 revolution per *sidereal month* of 27.321 662 days, where a sidereal month is measured with respect to the fixed stars. This is a reflection of the model having its coordinate system fixed in space (mean of J2000 equatorial coordinates).

Motion of the Apsides

A comparison of the model of Eqs. (1–3) with the new model of Eqs. (17–19) shows that the new model includes an important term that does not appear in the conventional model, having a frequency of about 70.99 rad cy^{-1}. This frequency reflects the motion of the line of apsides of the lunar orbit. The expected frequency of this motion may be computed from the periods of the anomalistic and sidereal months:

$$\begin{aligned} \omega &= \frac{2\pi}{\text{sidereal mo.}} - \frac{2\pi}{\text{anomalistic mo.}} \\ &= \left(\frac{2\pi}{27.321\,662^{\mathrm{d}}} - \frac{2\pi}{27.554\,550^{\mathrm{d}}} \right) \times 36\,525 \ \frac{\text{days}}{\text{cy}} \\ &= 70.9932 \text{ rad cy}^{-1} \end{aligned} \qquad (20)$$

in close agreement with the frequencies found using the curve fit.

ERROR ANALYSIS

The results shown in Eqs. (17–19) have been checked against the DE200 ephemeris model by using DE200 to generate lunar X, Y, and Z coordinates at 2^{20} (over one million) time points between A.D. 2000 January 1 and A.D. 2100 January 1, corresponding to roughly one point every fifty minutes for 100 years. The model shown in Eqs. (17–19) was run at the same time points, and the results compared with the DE200 results. This error analysis shows an rms position error between DE200 and the new model of Eqs. (17–19) of 0$''$341, and a maximum error of 1$''$033.

CONCLUSIONS

Three lunar ephemeris models for on-board flight software use have been discussed. A modified two-body model is very fast, but is of low precision and requires constant maintenance in the form of periodic updates of orbital elements from the ground. The model currently in common use, which is based on the low-precision formulae in the *AstronomicalAlmanac*, is of very good precision and will run indefinitely without ground intervention, but requires code to convert the calculated ecliptic mean-of-date coordinates to equatorial J2000 Cartesian coordinates. The method developed in this paper is of intermediate precision, requires the least code of the three, and will also run indefinitely without ground intervention. It may have applications for small missions where computer resources are limited and its precision is acceptable.

REFERENCES

1. F.R. Moulton, *An Introduction to Celestial Mechanics* (Dover Publications, New York, 1970), p. 363.

2. G.W. Hill, "Researches in the Lunar Theory," *Am. J. Math.*, **1**, pp. 5 26 (1878).

3. Ibid., pp. 129 147.

4. Ibid., pp. 245–261.

5. G.W. Hill, "On the Part of the Motion of the Lunar Perigee Which Is a Function of the Mean Motions of the Sun and Moon," *Acta Math.*, **8**, pp. 1 -36 (1886).

6. E.W. Brown, *An Introductory Treatise on the Lunar Theory* (Cambridge University Press, Cambridge, 1896).

7. D. Boccaletti, G. Pucacco, *Theory of Orbits, Volume 1: Integrable Systems and Non-perturbative Methods* (Springer, Berlin, 1996), pp. 293 299.

8. J.M.A. Danby, *Fundamentals of Celestial Mechanics (2nd ed.)* (Willmann-Bell, Richmond, 1988), pp. 371 388.

9. J. Meeus, *Astronomical Algorithms* (Willmann-Bell, Richmond, 1991), pp. 307–317.

10. M. Chapront-Touzé, J. Chapront, *Lunar Tables and Programs from 4000 B.C. to A.D. 8000* (Willmann-Bell, Richmond, 1991).

11. F.R. Moulton, *An Introduction to Celestial Mechanics* (Dover, New York, 1970), pp. 337 -365.

12. N.E. Howard, *The Telescope Handbook and Star Atlas* (Crowell, New York, 1967), p. 4.

13. P.K. Seidelmann (ed.), *Explanatory Supplement to the Astronomical Almanac* (University Science Books, Mill Valley, California, 1992), pp. 128 130.

14. S.W. McCuskey, *Introduction to Celestial Mechanics* (Addison-Wesley, Reading, Massachusetts, 1963), pp. 32-69.

15. *The Astronomical Almanac for the Year 1999* (U. S. Government Printing Office, Washington, 1998), p. D46.

16. B. Emerson, *Nautical Almanac Office Technical Note Number 48: Approximate Lunar Coordinates* (Science and Engineering Research Council, Royal Greenwich Observatory, June 1979).

17. *Improved Lunar Ephemeris, 1952-1959* (U. S. Government Printing Office, Washington, 1954).

18. W.J. Eckert, M.J. Walker, D. Eckert, "Transformations of the Lunar Coordinates and Orbital Parameters," *Astron. J.*, **71**, pp. 314 332 (1966).

19. *The Astronomical Almanac for the Year 1999* (U. S. Government Printing Office, Washington, 1998), p. K6.

20. Op. cit., p. B19.

21. Op. cit., p. K6.

22. E.M. Standish, "Orientation of the JPL Ephemerides, DE200/LE200, to the Dynamical Equinox of J2000," *Astron. Astrophys.*, **114**, pp. 297 302 (1982).

23. E.M. Standish, "The observational basis for JPL's DE200, the planetary ephemerides of the Astronomical Almanac," *Astron. Astrophys.*, **233**, pp. 252 271 (1990).

COMPUTATIONAL OPTIMAL CONTROL BY SPECTRAL COLLOCATION WITH DIFFERENTIAL INCLUSION

Fariba Fahroo
Department of Mathematics, Code MA/Ff
ffahroo@nps.navy.mil

I. Michael Ross
Department of Aero/Astro, Code AA/Ro
Naval Postgraduate School, Monterey, CA 93943
imross@nps.navy.mil

ABSTRACT

In most direct methods for numerically solving optimal control problems, a standard collocation technique is used to parametrize the time history of the states and controls. This method reduces the original infinite dimensional optimization problem to a finite dimensional parameter optimization or a nonlinear programming problem (NLP). To reduce the computational complexity and size of these problems, the method of differential inclusion which eliminates the controls from the formulation has been proposed. It has been suggested that higher order quadrature rules are incompatible with the differential inclusion concept. In this paper we show that a pseudospectral collocation method overcomes this drawback in a unique manner. In order to show the effectiveness of this method as opposed to a direct method based on collocation techniques, we first describe the Legendre pseudospectral method. This method relies on the structure of orthogonal polynomials and can be easily adapted for the use in a collocation or a differential inclusion method. We present the simple cart problem and the Goddard problem by the two formulations and compare the results in each example.

INRODUCTION

There are two major categories for numerical solution of optimal control problems: Indirect and direct methods, [1, 2]. The indirect methods involve solving the necessary conditions (costate equations) derived from the Minimum Principle and essentially solve a two point boundary value problem (TPBVP). The solution to these problems require the hard task of finding an initial guess for a non-physical quantity, the costate variable, and even in the cases where a good guess is available, the radius of convergence for these methods is rather small; therefore, convergence to a solution is not easily obtained in most cases. To avoid some of the problems encountered in indirect methods, direct methods have become more popular in applications.

Direct methods can be basically described as solving the optimal control problem by discretizing it to a parameter optimization problem and then solving the resulting nonlinear programming problem (NLP). The conversion to a parameter optimization problem can be classified into two major categories: i) parametrization of the control variable only (as in the POST software), and ii) parametrization of both control and state variables (as in the OTIS program). In most direct methods, [2] the conversion to a parameter optimization problem is achieved by first dividing the time interval into a prescribed number of subintervals whose endpoints are called nodes. The unknowns are the value of the control and the states at these nodes, *the state and control parameters*. The cost function and the state equations can be expressed in terms of these parameters which effectively reduce the optimal control problem to an NLP that can be solved by a standard nonlinear programming code. The time histories of both the control and the state variables can be obtained by using an interpolation scheme. In the popular OTIS collocation scheme, for example, cubic splines are used as the interpolating polynomials over the time segments, [3]. To impose the state differential equations, the Simpson-Hermite implicit integration scheme is used. In other collocation schemes used in the works by Conway *et al.* higher order Gauss-Lobatto [4, 5] or Runge-Kutta type quadrature rules are used in the collocation scheme [6]. The use of higher order integration rules in a collocation method allows the user a higher order of accuracy with a bigger step size. The larger step size results in a smaller number of discretization nodes or optimization variables. Since the efficiency and even convergence of NLP problems improves for a smaller size problem, finding ways to *accurately* and *efficiently* discretize optimal control problems is of great interest in this area of research.

Recently, Seywald and Kumar [7, 8] have proposed the method of differential inclusion to eliminate the bounded controls from the formulation. When applicable, this idea simplifies the existing equations greatly and reduces the size and complexity of computations in the discretized version. Conway and Larson [5] have re-examined this method and its claims, and compare it to the use of higher order quadrature rules in collocation schemes. By comparing the results for several examples for the two discretization methods, they conclude that even in the limited cases where the differential inclusion is applicable (such as linear controls), the reduction in the size of NLP variable from the elimination of the controls is offset by the reduction in the accuracy in the method. In other words, for differential inclusion to obtain the same degree of accuracy as in the higher order quadrature rules, more nodes should be used. The higher number of nodes increases the NLP variables more than the reduction of the number of NLP variables obtained from the elimination of control variables. In Ref. [5] this problem with accuracy for the differential inclusion method was attributed to the use of Euler integration rule for the approximation of the state equation. This explicit integration rule is easy to use and is yet among the least accurate integration rules. The use of higher order rules result in implicit integration that makes it impossible to explicitly express the state derivatives at the nodes in terms of the discrete states.

Intrigued by the claims of this paper by Conway and Larson, we decided to revisit the claims of both methods by using a *spectral collocation* scheme which has all the advantages of an explicit integration scheme while allowing a formulation of differential inclusion. It also has the desired accuracy of higher order quadrature rules. In fact, for smooth problems, this method has spectral accuracy [9].

This pseudospectral collocation scheme uses orthogonal polynomials such as Legendre and Chebyshev polynomials for approximation of control and state variables. In this manner it differs from the existing collocation methods which use piecewise polynomials such as linear or cubic splines. These orthogonal polynomials are used extensively in spectral methods for solving fluid dynamics problems [9, 10], but their use in solving optimal control problems has created a new way of transforming these problems to NLP problems. One particular merit of the use of orthogonal polynomials is their close relationship to Gauss-type integration rules. This relationship can be exploited to derive simple rules for transforming the original optimal control problem to a system of algebraic equations. The efficiency and simplicity of these rules are best demonstrated in the Spectral Collocation method that Elnagar *et al.* [11, 12] have recently employed to solve a general class of optimal control problems. In their method (which is the one used in this paper), polynomial approximations of the state and control variables are considered where Lagrange polynomials are the trial functions and the unknown coefficients are the values of the state and control variables at the Legendre-Gauss-Lobatto (LGL) points. With this choice of collocation points and properties of the Lagrange polynomials, the state equations and the state and control constraints are readily transformed to algebraic equations. The state differential constraints are imposed by collocating the functions at the LGL points and using a differentiation matrix which is obtained by taking the analytic derivative of the interpolating polynomials and collocating them at the LGL points. In this sense, this method of imposing the state equations is in marked contrast to the numerical integration techniques that are used to approximate the differential equations in other collocation schemes.

PROBLEM FORMULATION

Consider the following optimal control problem. Determine the control function $\mathbf{u}(\tau)$, and the corresponding state trajectory $\mathbf{x}(\tau)$, that minimize the Bolza cost function:

$$\mathcal{J}(\mathbf{u},\mathbf{x},\tau_f) = \mathcal{M}[\mathbf{x}(\tau_f),\tau_f] + \int_{\tau_0}^{\tau_f} \mathcal{L}(\mathbf{x},\mathbf{u})d\tau \tag{1}$$

with $\mathbf{x} \in R^n$ and $\mathbf{u} \in R^m$ subject to the state dynamics

$$\dot{\mathbf{x}}(\tau) = \mathbf{f}(\mathbf{x}(\tau),\mathbf{u}(\tau)), \qquad \tau \in [\tau_0,\tau_f] \tag{2}$$

and boundary conditions:

$$\begin{aligned} \boldsymbol{\psi}_0[\mathbf{x}(\tau_0),\tau_0] &= \mathbf{0}, & (3)\\ \boldsymbol{\psi}_f[\mathbf{x}(\tau_f),\tau_f] &= \mathbf{0}, & (4) \end{aligned}$$

where $\boldsymbol{\psi}_0 \in R^p$ with $p \leq n$ and $\boldsymbol{\psi}_f \in R^q$ with $q \leq n$. We consider an autonomous system since an extension to a non-autonomous system is straight forward.

The control inequality and equality constraints are formulated as

$$\begin{aligned} \mathbf{g}[\mathbf{x}(\tau),\mathbf{u}(\tau)] &\leq \mathbf{0} \qquad \mathbf{g} \in R^s, & (5)\\ \mathbf{h}[\mathbf{x}(\tau),\mathbf{u}(\tau)] &= \mathbf{0}, \qquad \mathbf{h} \in R^r, & (6) \end{aligned}$$

For the differential inclusion formulation, the bounds on the controls are used to generate bounds on the rate of change of the state variables. In problems where the control can be written explicity in terms of the states and their rate of change (for example in problems where the control appears linearly) then this transformation is straightforward. Theoretically, this formulation is based on the idea of a hodograph, (see [7]). For fixed states $\mathbf{x}$, the hodograph $S(\mathbf{x})$ is defined as the set of all possible state rates that can be obtained by varying the controls within their bounds. Therefore,

$$S(\mathbf{x}) = \{\dot{\mathbf{x}} = R^n \,|\, \dot{\mathbf{x}} = f(\mathbf{x},\mathbf{u}), \mathbf{u} \in \Omega\} \tag{7}$$

where $\Omega(\mathbf{x})$ is the set of all admissible controls $\mathbf{u} \in R^m$

$$\Omega(\mathbf{x}) = \{\mathbf{u} = R^m \,|\, \mathbf{h}(\mathbf{x},\mathbf{u}) = \mathbf{0},\, \mathbf{g}(\mathbf{x},\mathbf{u}) \leq \mathbf{0}\} \tag{8}$$

Differential inclusion is based on assuming that there are smooth functions $\mathbf{p}$ and $\mathbf{q}$ such that the hodograph can be rewritten as

$$S(\mathbf{x}) = \{\dot{\mathbf{x}} = R^n \,|\, \mathbf{p}(\dot{\mathbf{x}},\mathbf{x}) = \mathbf{0}, \mathbf{q}(\dot{\mathbf{x}},\mathbf{x}) \leq \mathbf{0}\} \tag{9}$$

Once this mapping ($\Omega \to S$) is obtained, then Eqs. (2), (5) and (6) can be replaced by

$$\begin{aligned} \mathbf{p}(\mathbf{x},\dot{\mathbf{x}}) &= \mathbf{0} & (10)\\ \mathbf{q}(\mathbf{x},\dot{\mathbf{x}}) &\leq \mathbf{0} & (11) \end{aligned}$$

Note that for the differential inclusion method to work, the controls must also be eliminated from the cost function as well. Hence, in general, the differential inclusion method cannot be applied *directly* to the Bolza problem. It implicitly requires that the Bolza problem be transformed to the Mayer problem in the usual manner of defining a state variable x_{n+1} whose dynamics are governed by $\dot{x}_{n+1} = \mathcal{L}(\mathbf{x}, \mathbf{u})$. This is a minor drawback of the differential inclusion method. Hereafter, when discussing the differential inclusion method, we will assume that the problem has been recast as a Mayer optimal control problem.

The advantage of using the differential inclusion method is that the controls have been eliminated in the formulation. In the discretized version of this formulation, the elimination of controls results in the reduction of the number of optimization variables. This in turn can result in increased numerical efficiency in solving the optimal control problems numerically by the direct methods. The disadvantage of differential inclusion is that many problems cannot be rewritten in the form required by Eq. (9).

THE LEGENDRE PSEUDOSPECTRAL METHOD

In this section, we present a Legendre pseudospectral method (Legendre spectral collocation method) (see [10, 9]) for solving the optimal control problem formulated in the preceding section. The basic idea of this method is to seek polynomial approximations for the state and control functions in terms of their values at the LGL points, [10, 11, 12]. The time derivative of the state vector, $\dot{\mathbf{x}}(\tau)$, is expressed in terms of the state vector $\mathbf{x}(\tau)$ at the collocation points by the use of a differentiation matrix. In this manner, the optimal control problem is transformed to an NLP problem for the value of the states and the controls at the nodes.

Although this method is presented in detail Refs. [11, 12, 13], here we provide some details for the purpose of completeness. Since the problem presented in the previous section is formulated over the time interval $[\tau_0, \tau_f]$, and the LGL points lie in the interval $[-1, 1]$, we use the following transformation to express the problem for $t \in [t_0, t_f] = [-1, 1]$:

$$\tau = \frac{(\tau_f - \tau_0)t + (\tau_f + \tau_0)}{2} \tag{12}$$

It follows that by using Eq. (12), expressions (1-5) can be replaced by

$$\mathcal{J}(\mathbf{x}(\cdot), \mathbf{u}(\cdot), \tau_f) = \mathcal{M}[\mathbf{x}(1), \tau_f] + \frac{\tau_f - \tau_0}{2} \int_{-1}^{1} \mathcal{L}[\mathbf{x}(t), \mathbf{u}(t)]dt \tag{13}$$

$$\dot{\mathbf{x}}(t) = (\frac{\tau_f - \tau_0}{2})\,[\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))], \tag{14}$$

$$\begin{aligned} \boldsymbol{\psi}_0(\mathbf{x}(-1), \tau_0) &= 0 && (15)\\ \boldsymbol{\psi}_f(\mathbf{x}(1), \tau_f) &= 0 && (16)\\ \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) &\leq 0 && (17)\\ \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) &= 0 && (18) \end{aligned}$$

Following Refs. [10, 11, 12], let $L_N(t)$ be the Legendre polynomial of degree N on the interval $[-1, 1]$. In the Legendre collocation approximation of (13)-(18), we use the LGL points, $t_l, l = 0, \ldots, N$ which are given by

$$t_0 = -1, \quad t_N = 1,$$

and for $1 \leq l \leq N-1$, t_l are the zeros of $\dot{L}_N$, the derivative of the Legendre polynomial, L_N. For approximating the continuous equations, we seek polynomial approximations of the form

$$\mathbf{x}^N(t) = \sum_{l=0}^{N} \mathbf{x}(t_l)\phi_l(t), \tag{19}$$

$$\mathbf{u}^N(t) = \sum_{l=0}^{N} \mathbf{u}(t_l)\phi_l(t), \tag{20}$$

where, for $l = 0, 1, ..., N$

$$\phi_l(t) = \frac{1}{N(N+1)L_N(t_l)} \frac{(t^2-1)\dot{L}_N(t)}{t - t_l},$$

are the Lagrange polynomials of order N. It can be shown that

$$\phi_l(t_k) = \delta_{lk} = \begin{cases} 1 & \text{if } l = k \\ 0 & \text{if } l \neq k \end{cases}$$

From this property of ϕ_l it follows that

$$\mathbf{x}^N(t_l) = \mathbf{x}(t_l),\ \mathbf{u}^N(t_l) = \mathbf{u}(t_l) \tag{21}$$

From Equation (21) one can see that the values of the approximate state and control functions at the collocation points are given exactly by the values of the continuous functions at these points.

To express the derivative $\dot{\mathbf{x}}^N(t)$ in terms of $\mathbf{x}^N(t)$ at the collocation points t_l, we differentiate (19) which results in a matrix multiplication of the following form [13]:

$$\dot{\mathbf{x}}^N(t_k) = \sum_{l=0}^{N} D_{kl}\mathbf{x}(t_l), \tag{22}$$

where D_{kl} are entries of the $(N+1) \times (N+1)$ differentiation matrix $\mathbf{D}$

$$\mathbf{D} := [D_{kl}] := \begin{cases} \frac{L_N(t_k)}{L_N(t_l)} \cdot \frac{1}{t_k - t_l} & k \neq l \\ -\frac{N(N+1)}{4} & k = l = 0 \\ \frac{N(N+1)}{4} & k = l = N \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

In addition, the integral in (13) is discretized by

$$\int_{-1}^{1} \mathcal{L}(\mathbf{x}^N, \mathbf{u}^N)dt \approx \sum_{k=0}^{N} \mathcal{L}(\mathbf{a}_k, \mathbf{b}_k)w_k \tag{24}$$

where w_k are the weights given by

$$w_k := \frac{2}{N(N+1)} \frac{1}{[L_N(t_k)]^2} \quad k = 0, 1, \ldots, N.$$

To facilitate the NLP formulation, we use the notation

$$\mathbf{a}_l := \mathbf{x}(t_l),\ \mathbf{b}_l := \mathbf{u}(t_l).$$

For the derivative of the state vector $\mathbf{x}(t)$, collocated at the points t_k, we rewrite Eq. (22)

$$\mathbf{c}_k = \dot{\mathbf{x}}^N(t_k) = \sum_{l=0}^{N} D_{kl}\mathbf{a}_l. \tag{25}$$

The state equations and the initial and terminal state conditions are discretized by first substituting (22)-(25) in (14) and collocating at the LGL nodes, t_k. This process results in the optimal control problem (13)-(18) discretized by by the following NLP: Find coefficients

$$\mathbf{a} = (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N),\ \mathbf{b} = (\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_N)$$

and possibly the final time τ_f to minimize

$$\mathcal{J}^N(\mathbf{a}, \mathbf{b}) = \frac{\tau_f - \tau_0}{2} \sum_{k=0}^{N} \mathcal{L}(\mathbf{a}_k, \mathbf{b}_k) w_k + \mathcal{M}(\mathbf{a}_N, \tau_f) \tag{26}$$

subject to

$$\begin{aligned}
\mathbf{A}_k(\mathbf{a}, \mathbf{b}) &= (\frac{\tau_f - \tau_0}{2})\mathbf{f}(\mathbf{a}_k, \mathbf{b}_k) - \mathbf{c}_k = 0, \quad k = 0, \ldots, N, & (27)\\
\mathbf{B}_k(\mathbf{a}, \mathbf{b}) &= \mathbf{g}(\mathbf{a}_k, \mathbf{b}_k) \le 0, \quad k = 0, \ldots, N, & (28)\\
\mathbf{C}_k(\mathbf{a}, \mathbf{b}) &= \mathbf{h}(\mathbf{a}_k, \mathbf{b}_k) = 0, \quad k = 0, \ldots, N, & (29)\\
\boldsymbol{\psi}_0(\mathbf{a}_0, \tau_0) &= 0, & (30)\\
\boldsymbol{\psi}_f(\mathbf{a}_N, \tau_f) &= 0. & (31)
\end{aligned}$$

When possible, (i.e., a user obtaining an explicit mapping from Ω to S cf. Eq. (9)), the differential inclusion formulation of these discretized equation is straigthforward. The NLP reduces to finding $\mathbf{a} = (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N)$ and possibly the final time τ_f to minimize the transformed Mayer cost

$$\hat{\mathcal{J}}^N(\mathbf{a}) = \hat{\mathcal{M}}(\mathbf{a}_N, \tau_f) \tag{32}$$

subject to

$$\begin{aligned}
\hat{\mathbf{A}}_k(\mathbf{a}) &= \mathbf{p}(\mathbf{a}_k, \mathbf{c}_k) = 0 \quad k = 0, \ldots, N, & (33)\\
\hat{\mathbf{B}}_k(\mathbf{a}) &= \mathbf{q}(\mathbf{a}_k, \mathbf{c}_k) \le 0, \quad k = 0, \ldots, N, & (34)\\
\boldsymbol{\psi}_0(\mathbf{a}_0, \tau_0) &= 0, & (35)\\
\boldsymbol{\psi}_f(\mathbf{a}_N, \tau_f) &= 0. & (36)
\end{aligned}$$

The equations above show that in both the collocation and the differential inclusion formulations, the discretized equations preserve the structure of the continuous ones. By collocating the equations at the LGL points, and using the properties of the Lagrange polynomials, the functions are evaluated only at the LGL points without depending on the neighboring points. The derivative of the states is expressed in terms of the differentiation matrix (cf. Eq. (25)) which can be used in the differential inclusion formulation to express the state rates in terms of the state variables at the nodes. In this manner this method of discretization is quite different from the Euler integration rule used in Seywald's formulation:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \dot{\mathbf{x}}_i \Delta t_i, \tag{37}$$

or the higher order trapezoid rule:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + (\Delta t_i/2)[\dot{\mathbf{x}}_i + \dot{\mathbf{x}}_{i+1}] \tag{38}$$

As mentioned in Ref. [5], the more accurate integration rules such as trapezoid or Simpson are implicit integration rules which make expression of the state derivatives at the ith node in terms of the state variables impossible. With our formulation of the Legendre pseudospectral method, we have circumvented this difficulty and offer a method which is both accurate and adaptable to the differential inclusion formulation.

NUMERICAL EXAMPLES

Example 1: The Simple Cart Problem

As a first example, we consider the simple cart problem which has been considered as one of the test problems by Conway and Larson [5]. This problem has an analytic solution and has a linear control and a quadratic cost function with a fixed final time. The state variables are x_1, the displacement of the cart of unit mass, and x_2 the velocity, and the control u is the external force.

The equations of motion are

$$\begin{aligned} \dot{x}_1 &= x_2 && (39)\\ \dot{x}_2 &= -x_2 + u && (40) \end{aligned}$$

The cost function to be minimized is

$$J = \int_0^{t_f} u^2\,dt \tag{41}$$

The initial conditions are the rest conditions,

$$x_1(0) = 0, x_2(0) = 0$$

The final time condition is

$$\psi_f = x_1(t_f) - 2.694528 x_2(t_f) + 1.155356 = 0 \tag{42}$$

For $t_f = 2.0$ this problem has the following analytical optimal solutions:

$$\begin{aligned} u(t) &= \frac{1}{4}e^t - \frac{1}{2} && (43)\\ x_1(t) &= -\frac{3}{8}e^{-t} + \frac{1}{8}e^t - \frac{1}{2}t + \frac{1}{4} && (44)\\ x_2(t) &= \frac{3}{8}e^{-t} + \frac{1}{8}e^t - \frac{1}{2} && (45) \end{aligned}$$

The optimal cost function has the value

$$J = \int_0^2 u^2\,dt = 0.577678 \tag{46}$$

In [5] the problem is solved by both the Simpson collocation rule and the differential inclusion method for $\hat{N}$ time segments, $\hat{N} = 5, 10, 20$. For $\hat{N}$ segments, there are $N = \hat{N} + 1$ nodes. In the

N	Method	J_i	$\|J_i - J_{ana}\|$	N_p	Execution Time (Seconds)
	Analytic solution	0.577678			
6	Collocation (Simpson)	0.577668	0.00001	18	N/A
6	Diff. Inc. (Euler)	0.582800	0.005122	12	N/A
6	Collocation (LGL)	0.577678	8.34491e-07	18	3.63
6	Diff. Inc. (LGL)	0.582800	1.1770142e-06	12	2.13
11	Collocation (Simpson)	0.577678	0.00000	33	N/A
11	Diff. Inc. (Euler)	0.578935	0.001257	22	N/A
11	Collocation (LGL)	0.577677	1.028268e-07	33	12.42
11	Diff. Inc. (LGL)	0.577677	1.372542e-07	22	9.34
21	Collocation (Simpson)	0.577682	0.000004	63	N/A
21	Diff. Inc. (Euler)	0.577990	0.000312	42	N/A
21	Collocation (LGL)	0.57767798	1.546971e-08	63	51
21	Diff. Inc. (LGL)	0.5776780	6.8009795e-09	42	55

Table 1: Comparison of the Collocation and Differential Inclusion Methods

Simpson's rule the number of optimization parameters N_p for n states and m controls is $n \times N + m \times N$, while for the differential inclusion formulation where the m controls are eliminated, $N_p = n \times N$. The same number of optimization parameters are used for the LGL formulation of these methods.

As indicated earlier, it is necessary to recast this problem in a Mayer format for the application of the differential inclusion method. In Tables 1 and 2, we compare our results with those of Ref. [5]. As one can see, for all N the results for both the collocation and the differential inclusion based on the LGL discretization are more accurate than the results for Simpson or the Euler formulations. It appears that the LGL collocation method is the most accurate method and compared to the LGL differential inclusion offers more accuracy for fewer number of nodes. This observation is in line with the comparison of the Simpson and Euler differential inclusions. But in the LGL discretization the error is a lot smaller in both formulations and both are better than the results reported in [5]. In general, the error in the cost function decreases as N increases in all these problems.

All the examples were run with the random initial guesses, and the run-time for the problems show that the LGL implementation of the differential inclusion method ran with somewhat faster speeds when compared to the LGL collocation method with comparable accuracy.

N	Method	$u(t_f)$	$x_1(t_f)$	$x_2(t_f)$
	Analytic solution	1.347264	0.122881	0.474383
6	Collocation (Simpson)	1.326334	0.122749	0.474333
6	Diff. Inc. (Euler)	N/A	0.131702	0.477656
6	Collocation (LGL)	1.344735	0.122881	0.474382
6	Diff. Inc. (LGL)	N/A	0.123095	0.474461
11	Collocation (Simpson)	1.342595	0.122815	0.474358
11	Diff. Inc. (Euler)	N/A	0.125050	0.475188
11	Collocation (LGL)	1.346640	0.122911	0.474393
11	Diff. Inc. (LGL)	N/A	0.122839	0.474367
21	Collocation (Simpson)	1.346748	0.122868	0.474377
21	Diff. Inc. (Euler)	N/A	0.123432	0.474587
21	Collocation (LGL)	1.346748	0.122868	0.474377
21	Diff. Inc. (LGL)	N/A	0.123122	0.474472

Table 2: Comparison of final states for the Simpson , Euler Differential Inclusion and LGL Methods

Example 2: Goddard Problem

The second example is the Goddard problem which is to determine the optimal trajectory for maximizing the final altitude for a rocket ascending vertically. The acting forces are the inverse-square gravitational force and the atmospheric drag.

The variables are: the radial distance r, the velocity v, and mass m. The Thrust magnitude T is the control variable and is within the fixed bounds $0 \leq T \leq T_{max}$. The state constraint is a dynamic pressure limit $q \leq q_{max}$, with $q = \frac{1}{2}\rho v^2$, where $\rho = \rho_0 e^{\beta(1-r)}$ is the atmospheric density. The control problem is formulated as finding T to maximize $r(\tau_f)$ subject to the equations of motion

$$\frac{dr}{d\tau} = v, \tag{47}$$

$$\frac{dv}{d\tau} = \frac{T-D}{m} - \frac{\mu}{r^2},$$

$$\frac{dm}{d\tau} = -\frac{T}{c}. \tag{48}$$

where the control T is subject to the constraint

$$0 \leq T \leq T_{max} \tag{49}$$

The aerodynamic drag D is given by

$$D = qC_DA \tag{50}$$

In nondimensional units, the boundary conditions are

$$r(0) = 1.0, \quad v(0) = 0, m(0) = 1.0 \tag{51}$$

$$v(\tau_f) = \text{ free }, \quad m(\tau_f) = m_f = 0.6 \tag{52}$$

The state inequality constraint is

$$\frac{1}{2}\rho_0 e^{\beta(1-r)} v^2 - q_{max} \leq 0$$

The nondimensional values used in calculation are the ones used in [7]:

$$C_D = 0.05, \, (\rho_0 A) = 12,400, \, \beta = 500, \, c = 0.5, \, T_{max} = 3.5, \, \mu = 1.$$

The Differential Inclusion Formulation

The differential inclusion formulation is based on eliminating the controls from the state equations by using the equality and inequality constraints on the controls. In this problem, equation (48) defines T the control in terms of m the mass, i.e.,

$$T = -c\frac{dm}{d\tau} = -c\dot{m}.$$

Using this and Eq. (49) the control can be eliminated resulting in the following constraints

$$\frac{dr}{d\tau} - v \; = \; 0, \tag{53}$$

$$\frac{dv}{d\tau} + \frac{c\dot{m} + D}{m} + \frac{1}{r^2} \; = \; 0, \tag{54}$$

$$-\frac{dm}{d\tau} - \frac{T_{max}}{c} \; \leq \; 0, \tag{55}$$

$$-\frac{dm}{d\tau} \; \leq \; 0, \tag{56}$$

The initial conditions, final time conditions and the state inequality constraints remain the same as before. In a direct comparison to Seywald's results, Figures (1)-(4) show our results for $q_{max} = \infty$ while Figures (5)-(8) display the same for $q_{max} = 10$. It is clear that the collocation (CO) and differential inclusion (DI) methods show no noticeable difference in the results for the same number of nodes. As demonstrated for $N = 11$, the differential inclusion and collocation results are indistinguishable. What does make the difference, however is the computer run time. In all cases , the (DI) method runs much faster than the (CO) method. The run times are displayed in Table 3.

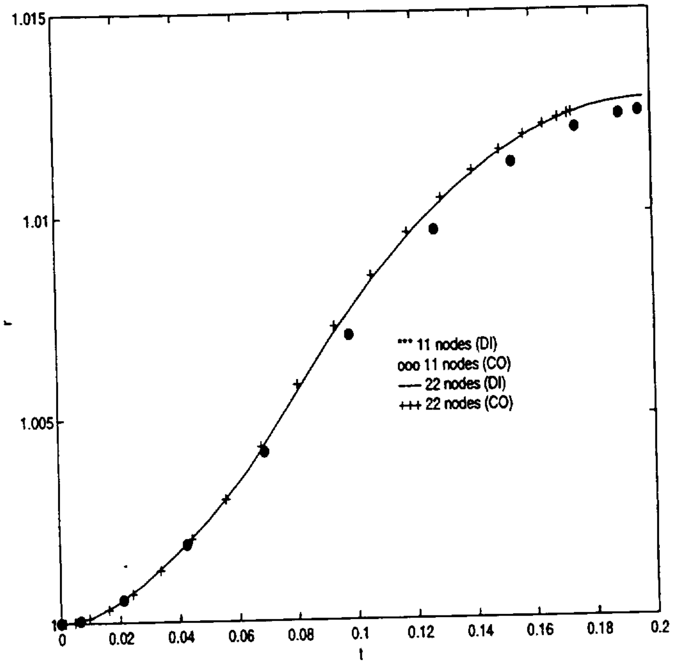


Figure 1: Time history of r for $q_{max} = \infty$ and $N = 11,\ 22$

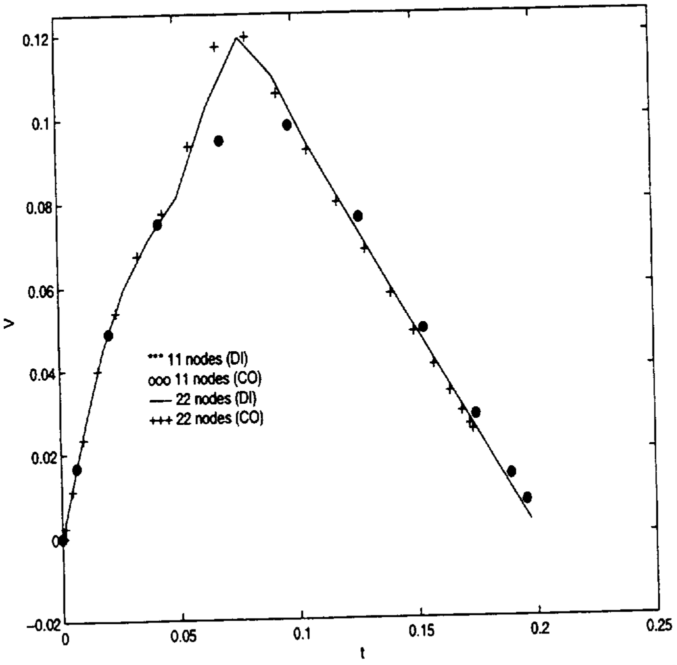


Figure 2: Time history of v for $q_{max} = \infty$ and $N = 11,\ 22$

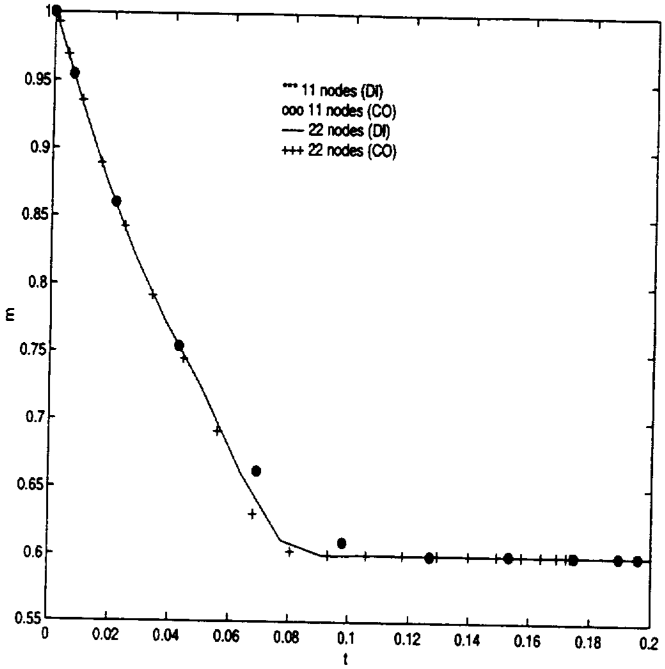


Figure 3: Time history of m for $q_{max} = \infty$ and $N = 11,\ 22$

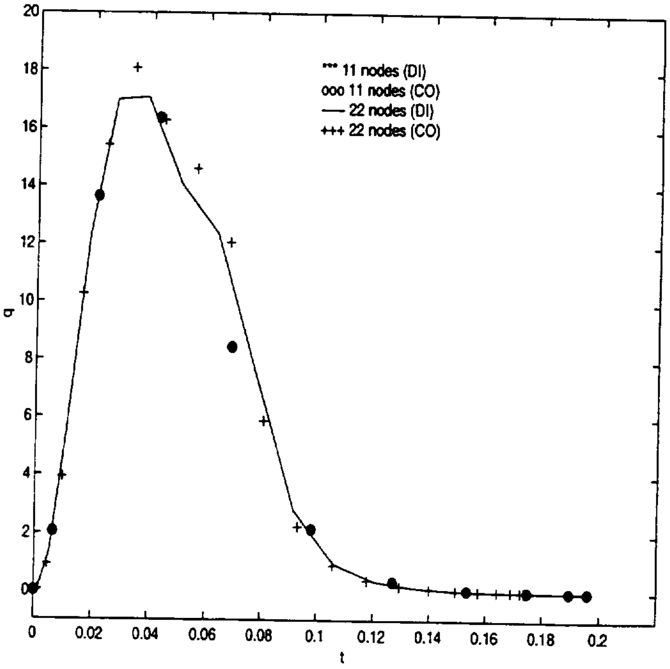


Figure 4: Time history of q for $q_{max} = \infty$ and $N = 11,\ 22$

The following set of computation was carried out with the constrained q where $q \leq q_{max} = 10$.

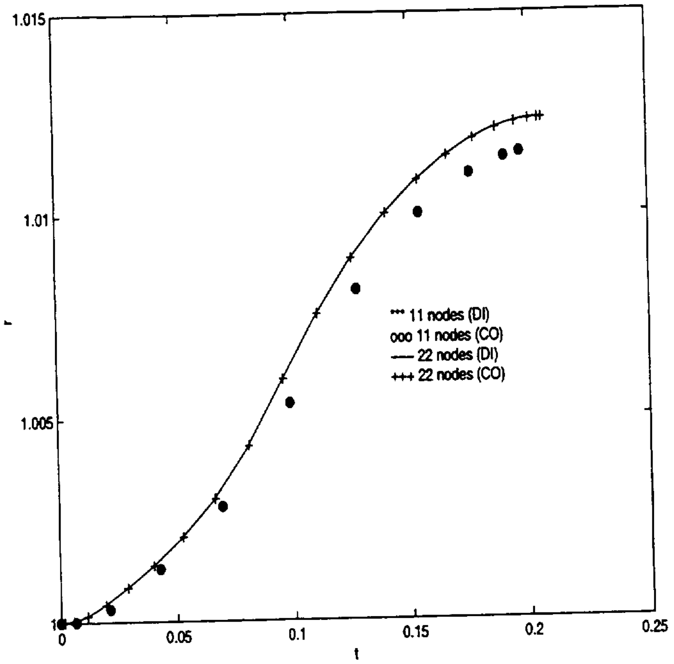


Figure 5: Time history of r for $q_{max} = 10$ and $N = 11,\ 22$

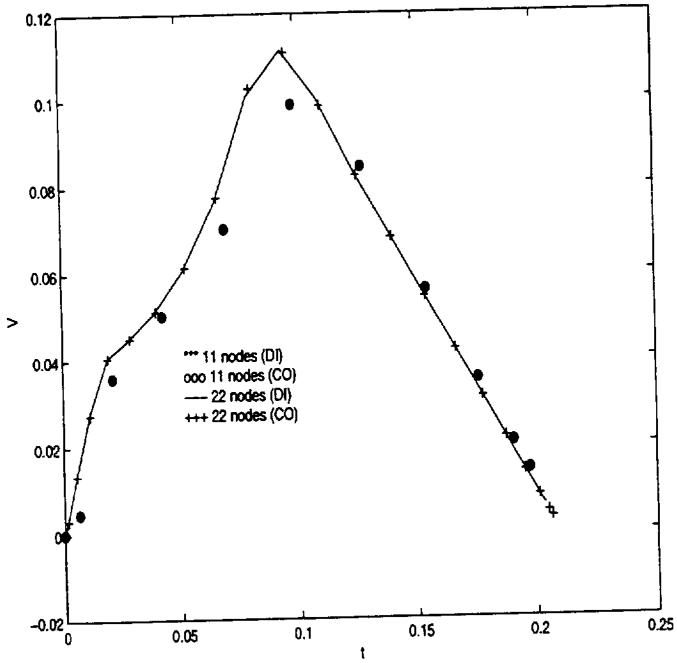


Figure 6: Time history of v for $q_{max} = 10$ and $N = 11,\ 22$

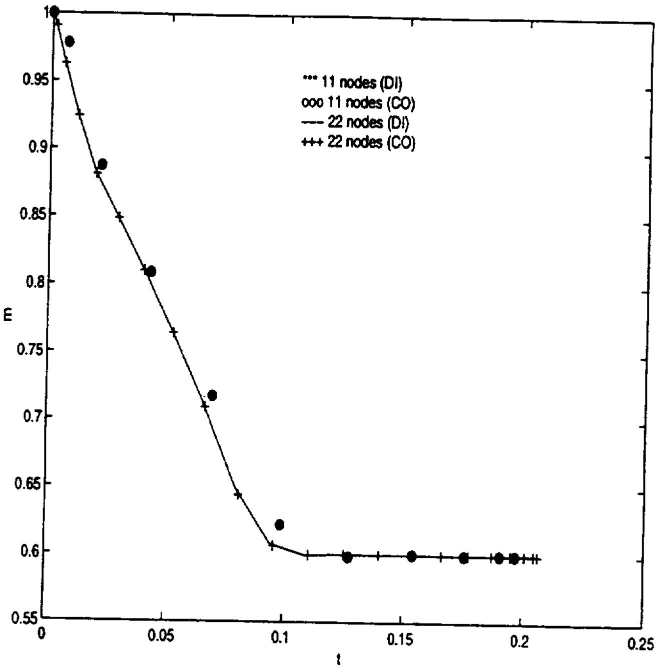


Figure 7: Time history of m for $q_{max} = 10$ and $N = 11, 22$

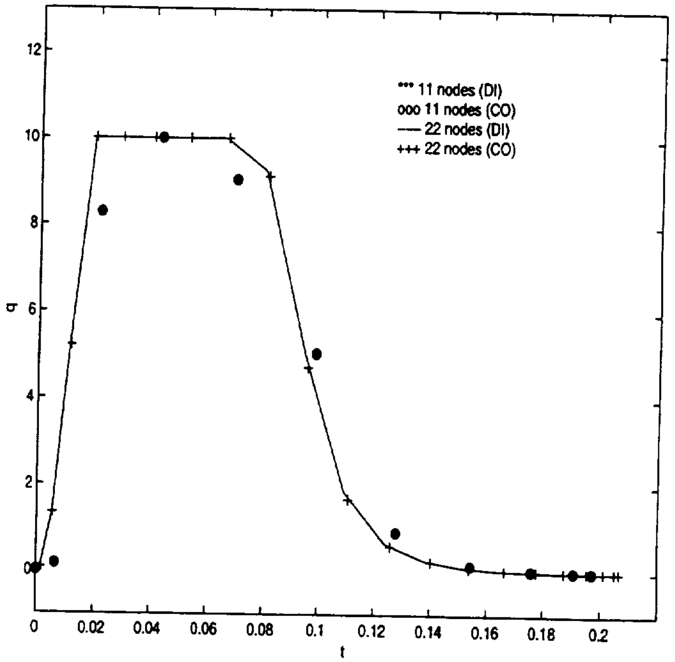


Figure 8: Time history of q for $q_{max} = 10$ and $N = 11, 22$

Computer Run Times		
N	(CO) Method Run Time	(DI) Method Run Time
5	89.40 sec	14.05 sec
10	194.5 sec	30.58 sec
20	2400 sec	301.5 sec

Table 3: Comparison of Run Times

CONCLUSIONS

The crux of the problem is in the implementation of the state dynamics equations. In direct collocation, the state equations are implemented as equality constraints whereas in the differential inclusion approach, they assume the form of both inequality and equality constraints. For the differential inclusion method to work, the value of the rate of change of state variables at the ith node should be expressible in terms of the discrete states. This limits the scope of the discretized differential inclusion method to simple Euler integration rules. Consequently, the gains obtained in reducing the size of the problem in differential inclusion method are lost due to the use of the less accurate Euler rule which requires more nodes to maintain acceptable accuracy. The pseudospectral collocation method presented in this paper overcomes these drawbacks since the calculation of the state derivatives in this method allows for expressing the derivative at the ith node as a linear combination of the discrete nodes. In this manner, the discretization of the derivative of the states is significantly different from the integration rules used in other collocation methods. The use of this highly accurate spectral collocation method in the discretization of the problem, makes the differential inclusion method quite competitive to direct collocation methods that employ the high order quadrature rules.

References

[1] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193-207.

[2] Hull, D. G., "Conversion of Optimal Control Problems into Parameter Optimization Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 57-60.

[3] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338-342.

[4] Herman, A. L., and Conway B.A., "Direct Optimization Using Collocation Based on High Order Gauss-Lobatto Quadrature Rules," *Journal of Guidance. Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592-599.

[5] Conway B.A. and Larson, K. M., "Collocation Versus Differential Inclusion in Direct Optimization," *Journal of Guidance. Control, and Dynamics*, Vol. 21, No. 5, 1998, pp. 780-785.

[6] Enright, P. J., and Conway B.A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance. Control, and Dynamics,* Vol. 15, No. 3, 1992, pp. 994-1002.

[7] Seywald, H., "Trajectory Optimization Based on Differential Inclusion," *Journal of Guidance. Control, and Dynamics,* Vol. 17, No. 3, 1994, pp. 480-487.

[8] Kumar, R. R. and Seywald, H., "Should Controls Be Eliminated While Solving Optimal Control Problems via Direct Method?," *Journal of Guidance. Control, and Dynamics,* Vol. 19, No. 2, 1996, pp. 418-423.

[9] Canuto, C., Hussaini, M.Y., Quarteroni, A., and Zang, T.A., *Spectral Methods in Fluid Dynamics,* Springer Verlag, New York, 1988.

[10] Gottlieb, D., Hussaini, M.Y., and Orszag, "Theory and Applications of Spectral Methods," in *Spectral Methods for PDE's,* Philadelphia, PA: SIAM, 1984.

[11] Elnagar, J., Kazemi, M.A. and Razzaghi, M., " The Pseudospectral Legendre Method for Discretizing Optimal Control Problem," *IEEE Transactions on Automatic Control,* Vol. 40, No. 10, 1995, pp. 1793-1796.

[12] Elnagar, J. and Razzaghi, M. "A Collocation-Type Method for Linear Quadratic Optimal Control Problems," *Optimal Control Applications and Methods,* Vol. 18, 1997, pp. 227-235.

[13] Fahroo, F. and Ross, I.M., "Costate Estimation by a Legendre Pseudospectral Method," *Proceedings of the 1998 AIAA GNC Conference,* Boston, MA, August 1998, AIAA Paper 98-4222.

OPTIMAL VARIABLE-STRUCTURE CONTROL TRACKING OF SPACECRAFT MANEUVERS

John L. Crassidis, Srinivas R. Vadali
Texas A&M University
Department of Aerospace Engineering
College Station, TX 77843-3141

F. Landis Markley
NASA-Goddard Space Flight Center
Guidance, Navigation, and Control Center, Code 571
Greenbelt, MD 20771

ABSTRACT

An optimal control approach using variable-structure (sliding-mode) tracking for large angle spacecraft maneuvers is presented. The approach expands upon a previously derived regulation result using a quaternion parameterization for the kinematic equations of motion. This parameterization is used since it is free of singularities. The main contribution of this paper is the utilization of a simple term in the control law that produces a maneuver to the reference attitude trajectory in the shortest distance. Also, a multiplicative error quaternion between the desired and actual attitude is used to derive the control law. Sliding-mode switching surfaces are derived using an optimal-control analysis. Control laws are given using either external torque commands or reaction wheel commands. Global asymptotic stability is shown for both cases using a Lyapunov analysis. Simulation results are shown which use the new control strategy to stabilize the motion of the Microwave Anisotropy Probe spacecraft.

INTRODUCTION

The control of spacecraft for large angle slewing maneuvers poses a difficult problem. Some of these difficulties include: the highly nonlinear characteristics of the governing equations, control rate and saturation constraints and limits, and incomplete state knowledge due to sensor failure or omission. The control of spacecraft with large angle slews can be accomplished by either open-loop or closed-loop schemes. Open-loop schemes usually require a pre-determined pointing maneuver and are typically determined using optimal control techniques, which involve the solution of a two-point boundary value problem (e.g., the time optimal maneuver problem[1]). Also, open-loop schemes are sensitive to spacecraft parameter uncertainties and unexpected disturbances.[2] Closed-loop systems can account for parameter uncertainties and disturbances, and thus provide a more robust design methodology.

In recent years, much effort has been devoted to the closed-loop design of spacecraft with large angle slews. Wie and Barba[3] derive a number of simple control schemes using quaternion and angular velocity (rate) feedback. Other full state feedback techniques have been developed that are based on variable-structure (sliding-mode) control, which uses a feedback linearizing technique and an additional term aimed at dealing with model uncertainty.[4] A variable-structure controller has been developed for the regulation of spacecraft maneuvers using a Gibbs vector parameterization,[5] a modified-Rodrigues parameterization,[6] and a quaternion parameterization.[7] In both [3] and [7], a term was added so that the spacecraft maneuver follows the shortest path and requires the least amount of control torque. The variable-structure control approach using a quaternion parameterization has been recently expanded to the attitude tracking case.[8,9] However, these controllers do not take into account the shortest possible path as shown in Refs. [3] and [7]. This paper expands upon the results in Ref. [7] to provide an optimal control law for asymptotic tracking of spacecraft maneuvers using variable-structure control.

The organization of this paper proceeds as follows. First, the kinematic and dynamic equations of motion are summarized. Then, an analysis for the selection of the switching surfaces is shown. This is shown using both an optimal control approach and a Lyapunov stability-based approach. Also, variable-structure considerations will be investigated. Next, a variable-structure controller is derived using either external torque inputs or reaction wheels. Finally, simulation results will be shown which use the new control law to stabilize the motion of the Microwave Anisotropy Probe spacecraft.

BACKGROUND

In this section, a brief review of the kinematic and dynamic equations of motion for a three-axis stabilized spacecraft is shown. The attitude is assumed to be represented by the quaternion, defined as[10]

$$\boldsymbol{q} \equiv \begin{bmatrix} \boldsymbol{q}_{13} \\ q_4 \end{bmatrix} \tag{1}$$

with

$$\boldsymbol{q}_{13} \equiv \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \hat{\boldsymbol{n}} \sin(\Phi/2) \tag{2a}$$

$$q_4 = \cos(\Phi/2) \tag{2b}$$

where $\hat{\boldsymbol{n}}$ is a unit vector corresponding to the axis of rotation and Φ is the angle of rotation. The quaternion kinematic equations of motion are derived by using the spacecraft's angular velocity ($\boldsymbol{\omega}$), given by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\Omega(\boldsymbol{\omega})\boldsymbol{q} = \frac{1}{2}\Xi(\boldsymbol{q})\boldsymbol{\omega} \tag{3}$$

where $\Omega(\boldsymbol{\omega})$ and $\Xi(\boldsymbol{q})$ are defined as

$$\Omega(\boldsymbol{\omega}) \equiv \begin{bmatrix} -[\boldsymbol{\omega}\times] & \vdots & \boldsymbol{\omega} \\ \cdots\cdots & \vdots & \cdots\cdots \\ -\boldsymbol{\omega}^T & \vdots & 0 \end{bmatrix} \tag{4a}$$

$$\Xi(\boldsymbol{q}) \equiv \begin{bmatrix} q_4 I_{3\times3} + [\boldsymbol{q}_{13}\times] \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ -\boldsymbol{q}_{13}^T \end{bmatrix} \tag{4b}$$

where $I_{n\times n}$ represents an $n\times n$ identity matrix (also, $0_{n\times m}$ will represent an $n\times m$ zero matrix). The 3×3 dimensional matrices $[\boldsymbol{\omega}\times]$ and $[\boldsymbol{q}_{13}\times]$ are referred to as cross product matrices since $\boldsymbol{a}\times\boldsymbol{b} = [\boldsymbol{a}\times]\boldsymbol{b}$, with

$$[\boldsymbol{a}\times] \equiv \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{5}$$

Since a three degree-of-freedom attitude system is represented by a four-dimensional vector, the quaternion components cannot be independent. This condition leads to the following normalization constraint

$$q^T q = q_{13}^T q_{13} + q_4^2 = 1 \tag{6}$$

The matrix $\Xi(q)$ obeys the following relations that are extremely useful

$$\Xi^T(q)\Xi(q) = q^T q \, I_{3\times 3} \tag{7a}$$

$$\Xi(q)\Xi^T(q) = q^T q \, I_{4\times 4} - q\, q^T \tag{7b}$$

$$\Xi^T(q) q = \mathbf{0}_{3\times 1} \tag{7c}$$

$$\Xi^T(q)\boldsymbol{\zeta} = -\Xi^T(\boldsymbol{\zeta}) q \quad \text{for any } \boldsymbol{\zeta}_{4\times 1} \tag{7d}$$

$$\Xi^T(\boldsymbol{\zeta})\Xi(q) = \left\{ \left(q^T \boldsymbol{\zeta}\right) I_{3\times 3} + \left[\Xi^T(\boldsymbol{\zeta}) q \times\right]\right\} \quad \text{for any } \boldsymbol{\zeta}_{4\times 1} \tag{7e}$$

Also, the error quaternion between two quaternions, q and q_d, is defined by

$$\delta q \equiv \begin{bmatrix} \delta q_{13} \\ \hdashline \delta q_4 \end{bmatrix} = q \otimes q_d^{-1} \tag{8}$$

where the operator ⊗ denotes quaternion multiplication (see Ref. [10] for details), and the inverse quaternion is defined by

$$q_d^{-1} = \begin{bmatrix} -q_{d_1} & -q_{d_2} & -q_{d_3} & q_{d_4} \end{bmatrix}^T \tag{9}$$

Other useful identities are given by

$$\delta q_{13} = \Xi^T(q_d) q \tag{10a}$$

$$\delta q_4 = q^T q_d \tag{10b}$$

Also, if Equation (8) represents a small rotation then $\delta q_4 \approx 1$, and δq_{13} corresponds to half-angles of rotation.

The dynamic equations of motion, also known as Euler's equations, for a rotating spacecraft with are given by[11]

$$J\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times (J\boldsymbol{\omega}) + \boldsymbol{u} \tag{11}$$

where J is the inertia matrix of the spacecraft, and $\boldsymbol{u}$ is the total external torque input. If the spacecraft is equipped with 3 orthogonal reaction or momentum wheels, then Euler's equations become:

$$\left(J - \bar{J}\right)\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \left(J\boldsymbol{\omega} + \bar{J}\bar{\boldsymbol{\omega}}\right) - \bar{\boldsymbol{u}} \tag{12a}$$

$$\bar{J}\left(\dot{\bar{\boldsymbol{\omega}}} + \dot{\boldsymbol{\omega}}\right) = \bar{\boldsymbol{u}} \tag{12b}$$

where $\bar{J}$ is the diagonal inertia matrix of the wheels, J now includes the mass of the wheels, $\bar{\boldsymbol{\omega}}$ is the wheel angular velocity vector, and $\bar{\boldsymbol{u}}$ is the wheel torque vector.

SELECTION OF SWITCHING SURFACES

Optimal Control Analysis

The variable-structure control design is used to track a desired quaternion $\boldsymbol{q}_d$ and corresponding angular velocity $\boldsymbol{\omega}_d$. As shown previously for regulation,[7] under ideal sliding conditions, the trajectory in the state-space moves on the sliding manifold. For tracking, the following loss function is minimized to determine the optimal switching surfaces:

$$\Pi(\boldsymbol{\omega}) = \frac{1}{2}\int_{t_s}^{\infty}\left[\rho\,\delta\boldsymbol{q}_{13}^T\delta\boldsymbol{q}_{13} + (\boldsymbol{\omega}-\boldsymbol{\omega}_d)^T(\boldsymbol{\omega}-\boldsymbol{\omega}_d)\right]dt \tag{13}$$

subject to the bilinear system constraint given in Equation (3). Note that ρ is a scalar gain and t_s is the time of arrival at the sliding manifold. Minimization of Equation (13) leads to the following two-point-boundary-value-problem:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\Xi(\boldsymbol{q})\boldsymbol{\omega} \tag{14a}$$

$$\dot{\boldsymbol{\lambda}} = -\rho\,\Xi(\boldsymbol{q}_d)\Xi^T(\boldsymbol{q}_d)\boldsymbol{q} + \frac{1}{2}\Xi(\boldsymbol{\lambda})\boldsymbol{\omega} \tag{14b}$$

$$\boldsymbol{\omega}-\boldsymbol{\omega}_d = -\frac{1}{2}\Xi^T(\boldsymbol{q})\boldsymbol{\lambda} \tag{14c}$$

where $\boldsymbol{\lambda}$ is the co-state vector. The following sliding vector is chosen:

$$\boldsymbol{s} \equiv (\boldsymbol{\omega}-\boldsymbol{\omega}_d) + k\,\Xi^T(\boldsymbol{q}_d)\boldsymbol{q} = \boldsymbol{0} \tag{15}$$

where k is a scalar gain. The sliding vector is optimal if the solution of Equation (15) minimizes Equation (13). This can be proven by first substituting Equation (15) into Equation (14c) and using the matrix identities in Equation (7), yielding

$$\boldsymbol{\lambda} = -2k\boldsymbol{q}_d \tag{16}$$

Next, using the fact that the desired quaternion can be obtained from the following

$$\dot{\boldsymbol{q}}_d = \frac{1}{2}\Xi(\boldsymbol{q}_d)\boldsymbol{\omega}_d \tag{17}$$

leads directly to

$$\dot{\boldsymbol{\lambda}} = -k\,\Xi(\boldsymbol{q}_d)\boldsymbol{\omega}_d \tag{18}$$

Comparing Equation (18) to Equation (14b), and using Equation (15) now leads to the following relationship:

$$-k\,\Xi(\boldsymbol{q}_d)\boldsymbol{\omega}_d = -\rho\,\Xi(\boldsymbol{q}_d)\Xi^T(\boldsymbol{q}_d)\boldsymbol{q} - k\,\Xi(\boldsymbol{q}_d)\boldsymbol{\omega}_d + k^2\Xi(\boldsymbol{q}_d)\Xi^T(\boldsymbol{q}_d)\boldsymbol{q} \tag{19}$$

Equation (19) is satisfied for $k = \pm\sqrt{\rho}$. Therefore, the sliding condition in Equation (15) leads to an optimal solution (i.e., minimum Π in Equation (13)).

For this special case, it can be shown that the value of the loss function in Equation (13) is given by

$$\Pi^* = 2k\left[1 - \delta q_4(t_s)\right] \tag{20}$$

where k must now be strictly positive. Note that δq_4 corresponds directly to the cosine of half the angle error of rotation. Both $\boldsymbol{\delta q}$ and $-\boldsymbol{\delta q}$ represent the same rotation; however, the value of the loss function in Equation (20) is significantly different for each rotation. One rotation ($\boldsymbol{\delta q}$) gives the shortest distance to the sliding manifold, while the other ($-\boldsymbol{\delta q}$) gives the longest distance. Although each rotation gives the same orientation, more energy may be required to maneuver the spacecraft using $-\boldsymbol{\delta q}$. In order to give the shortest possible distance the following sliding vector is chosen:

$$s \equiv (\boldsymbol{\omega} - \boldsymbol{\omega}_d) + k\,\text{sgn}\left[\delta q_4(t_s)\right]\Xi^T(q_d)q = \mathbf{0} \tag{21}$$

where it is assumed that $\delta q_4(t_s)$ is non-zero for a finite time. Using this sliding condition leads to the following value for the loss function:

$$\Pi^* = 2k\left[1 - \left|\delta q_4(t_s)\right|\right] \tag{22}$$

which yields a minimal value for any rotation. The kinematic equation for δq_4 can now be written as

$$\delta\dot{q}_4 = \frac{1}{2}k\left(1 - \delta q_4^2\right)\text{sign}\left[\delta q_4(t_s)\right] \tag{23}$$

Therefore, the derivative $\delta\dot{q}_4$ is either positive or negative depending on the sign of $\delta q_4(t_s)$, so the term $\text{sgn}\left[\delta q_4(t_s)\right]$ can be replaced with $\text{sgn}\left[\delta q_4(t - t_s)\right]$ without loss in generality (note, this corresponds to the time *after* the sliding manifold is reached).

Lyapunov Analysis

The sliding vector shown in Equation (21) can also be shown to be stable using a Lyapunov analysis. The time derivative of Equation (10a) can be shown to be given by

$$\boldsymbol{\delta\dot{q}}_{13} = \frac{1}{2}\delta q_4\,(\boldsymbol{\omega} - \boldsymbol{\omega}_d) + \frac{1}{2}[\boldsymbol{\delta q}_{13}\times](\boldsymbol{\omega} + \boldsymbol{\omega}_d) \tag{24}$$

Next, the following candidate Lyapunov function is chosen:

$$V_s = \frac{1}{2}\boldsymbol{\delta q}_{13}^T\boldsymbol{\delta q}_{13} \tag{25}$$

Using the sliding vector in Equation (21), the time derivative of Equation (25) is given by

$$\dot{V}_s = -\frac{1}{2}k\left|\delta q_4\right|\boldsymbol{\delta q}_{13}^T\boldsymbol{\delta q}_{13} \le 0 \tag{26}$$

Hence, V_s is indeed a Lyapunov function for $k > 0$. This analysis generalizes the results shown in Ref. [9], where the spacecraft's attitude is restricted in the workspace defined with $q_4 > 0$.

Sliding Manifold Considerations

The term $\text{sgn}\left[\delta q_4(t_s)\right]$ is used to develop a control law that yields the shortest distance to the sliding manifold. However, in actual practice this manifold is difficult to visualize. This section shows an analysis for the optimality conditions during the transient response of the control design. The goal of the controller is to drive $s \to \mathbf{0}$. Since the norm-squared of sliding vector is closely related to the energy of the system, this can be useful to determine the optimal maneuver conditions for the spacecraft controller. Consider the following sliding vector:

$$\mathbf{s} = (\boldsymbol{\omega} - \boldsymbol{\omega}_d) + \psi \Xi^T(\mathbf{q}_d)\mathbf{q} \tag{27}$$

where ψ is some scalar (note: the sliding vector is now assumed to be non-zero). The following performance measure can be used to quantify how fast the sliding mode manifold is reached:

$$\frac{1}{2}\int_{t_0}^{t_1} \mathbf{s}^T\mathbf{s}\, dt = \frac{1}{2}\int_{t_0}^{t_1}\left[(\boldsymbol{\omega}-\boldsymbol{\omega}_d)^T(\boldsymbol{\omega}-\boldsymbol{\omega}_d) + \psi^2 \delta\mathbf{q}_{13}^T\delta\mathbf{q}_{13} + 2\psi(\boldsymbol{\omega}-\boldsymbol{\omega}_d)^T\,\Xi^T(\mathbf{q}_d)\mathbf{q}\right]dt \tag{28}$$

where t_1 corresponds to the time to reach $\mathbf{s} = \mathbf{0}$. The integral in Equation (28) during the transient response should be evaluated in order to determine how various choices of ψ affect the transient response. However, a closed-form solution for the integral is extremely difficult to obtain in this case, since one now requires knowledge of the actual control inputs which affect the angular velocity and attitude. Still, a closed-form solution for the last term in Equation (28) can be found without this knowledge. The time derivative of Equation (10b) can be shown to be given by

$$\delta\dot{q}_4 = -\frac{1}{2}(\boldsymbol{\omega}-\boldsymbol{\omega}_d)^T\,\Xi^T(\mathbf{q}_d)\mathbf{q} \tag{29}$$

The last term in Equation (28) is now given by

$$\frac{1}{2}\int_{t_0}^{t_1}\left[2\psi(\boldsymbol{\omega}-\boldsymbol{\omega}_d)^T\,\Xi^T(\mathbf{q}_d)\mathbf{q}\right]dt = -2\psi\left[\delta q_4(t_1) - \delta q_4(t_0)\right] \tag{30}$$

Therefore, once a control has been determined then only the first two terms of the right-hand-side of Equation (28) need to be numerically determined. It will be shown through an analysis of the closed-form structure of Equation (29) that $\psi = k\,\mathrm{sgn}\left[\delta q_4(t)\right]$ should be used at all times (even before the sliding manifold is reached) in order to produce a response in the shortest distance. Equations (28) and (30) can be used to validate this approach from dynamic numerical simulations.

VARIABLE-STRUCTURE TRACKING

The goal of the variable-structure controller is to track a desired quaternion $\mathbf{q}_d$ and corresponding angular velocity $\boldsymbol{\omega}_d$. The variable-structure control design with external torques only is given by

$$\mathbf{u} = [\boldsymbol{\omega}\times]J\,\boldsymbol{\omega} + J\left\{\frac{1}{2}k\,\mathrm{sgn}(\delta q_4)\left[\Xi^T(\mathbf{q})\Xi(\mathbf{q}_d)\boldsymbol{\omega}_d - \Xi^T(\mathbf{q}_d)\Xi(\mathbf{q})\boldsymbol{\omega}\right] + \dot{\boldsymbol{\omega}}_d - G\,\boldsymbol{\vartheta}\right\} \tag{31}$$

where G is a 3×3 positive definite, diagonal matrix, and the i^{th} component of $\boldsymbol{\vartheta}$ is given by

$$\vartheta_i = \mathrm{sat}(s_i, \varepsilon_i), \qquad i = 1,2,3 \tag{32}$$

As stated previously, the term $\mathrm{sgn}(\delta q_4)$ is used to drive the system to the desired trajectory in the shortest distance. The saturation function is used to minimize chattering in the control torques. This function is defined by

$$\mathrm{sat}(s_i, \varepsilon_i) \equiv \begin{cases} 1 & \text{for} \quad s_i > \varepsilon_i \\ \dfrac{s_i}{\varepsilon} & \text{for} \quad |s_i| \le \varepsilon_i \\ -1 & \text{for} \quad s_i < -\varepsilon_i \end{cases} \qquad i = 1,2,3 \tag{33}$$

where ε is a small positive quantity. Also, the sliding manifold is given by

$$\mathbf{s} = (\boldsymbol{\omega} - \boldsymbol{\omega}_d) + k\,\mathrm{sgn}(\delta q_4)\,\Xi^T(\mathbf{q}_d)\,\mathbf{q} \tag{34}$$

The stability of the closed-loop system using this controller can be evaluated using the following candidate Lyapunov function

$$V = \frac{1}{2}\mathbf{s}^T\mathbf{s} \tag{35}$$

Using Equations (11), (31), and (34) the time-derivative of Equation (35) can be shown to be given by

$$\dot{V} = -\mathbf{s}^T G\boldsymbol{\vartheta} \tag{36}$$

which is always less than or equal to zero as long as G is positive definite.

If wheels are used to control the spacecraft, then the sliding mode controller is given by the following:

$$\bar{\mathbf{u}} = -[\boldsymbol{\omega}\times](J\boldsymbol{\omega} + \bar{J}\bar{\boldsymbol{\omega}}) + (J - \bar{J})\left\{\frac{1}{2}k\,\mathrm{sgn}(\delta q_4)\left[\Xi^T(\mathbf{q}_d)\Xi(\mathbf{q})\boldsymbol{\omega} - \Xi^T(\mathbf{q})\Xi(\mathbf{q}_d)\boldsymbol{\omega}_d\right] - \dot{\boldsymbol{\omega}}_d + G\boldsymbol{\vartheta}\right\} \tag{37}$$

The stability of the system can also be easily proven using the Lyapunov function in Equation (35).

Analysis

In this section an analysis of using $\mathrm{sgn}(\delta q_4)$ for all times in the control law is shown. We first assume that the desired angular velocity is zero $(\boldsymbol{\omega}_d = \mathbf{0})$ and that the matrix G is given by a scalar times the identity matrix $(g\,I_{3\times 3})$. We further assume that the thickness of the boundary layer ε and the gain g are sufficiently large so that

$$G\boldsymbol{\vartheta} = \beta\boldsymbol{\omega} + \beta k\,\mathrm{sgn}(\delta q_4)\,\boldsymbol{\delta q}_{13} \tag{38}$$

where $\beta = g/\varepsilon$. Using Equations (11), (24), and (31), the closed-loop dynamics for $\boldsymbol{\omega}$ now become

$$\dot{\boldsymbol{\omega}} = -\frac{1}{2}k\,\mathrm{sgn}(\delta q_4)\,\Xi^T(\mathbf{q}_d)\Xi(\mathbf{q})\boldsymbol{\omega} - \beta\boldsymbol{\omega} - \beta k\,\mathrm{sgn}(\delta q_4)\,\boldsymbol{\delta q}_{13} \tag{39}$$

Next, using Equation (7e) with $\boldsymbol{\zeta} = \mathbf{q}_d$ leads directly to

$$\dot{\boldsymbol{\omega}} = -\frac{1}{2}k\,\mathrm{sgn}(\delta q_4)\{\delta q_4 I_{3\times 3} + [\boldsymbol{\delta q}_{13}\times]\}\boldsymbol{\omega} - \beta\boldsymbol{\omega} - \beta k\,\mathrm{sgn}(\delta q_4)\,\boldsymbol{\delta q}_{13} \tag{40}$$

Taking the time derivative of Equation (29), and using both Equation (40) and the quaternion constraint equation yield

$$\delta\ddot{q}_4 + \left(\frac{1}{2}k|\delta q_4| + \beta\right)\delta\dot{q}_4 + \left(\frac{1}{2}\beta k|\delta q_4| + \frac{1}{4}\boldsymbol{\omega}^T\boldsymbol{\omega}\right)\delta q_4 = \frac{1}{2}\beta k\,\mathrm{sgn}(\delta q_4) \tag{41}$$

Equation (41) represents a second-order nonlinear spring-mass-damper type system with an exogenous step input. The stability of Equation (41) can be evaluated by considering the following candidate Lyapunov function

$$V_{\delta q_4} = \frac{1}{2}\delta\dot{q}_4^2 + \frac{1}{2}\left(\frac{1}{4}\boldsymbol{\omega}^T\boldsymbol{\omega}\right)\delta q_4^2 + \frac{1}{2}\beta k\left[1-\delta q_4 \operatorname{sgn}(\delta q_4)\right] \tag{42}$$

Also, the last term in Equation (41) is always greater or equal to zero since $\beta > 0$, $k > 0$, and $0 \le \delta q_4 \operatorname{sgn}(\delta q_4) \le 1$. Next, we assume that δq_4 is non-zero for a finite time. This is a valid assumption since the control law is known to produce asymptotic tracking in the closed-loop attitude response. Therefore, taking the time-derivative of Equation (42), and using Equations (40) and (41) give

$$\dot{V}_{\delta q_4} = -\left(\frac{1}{2}k\left|\delta q_4\right| + \beta\right)\delta\dot{q}_4^2 - \frac{1}{4}\left(\beta + \frac{1}{2}k\left|\delta q_4\right|\right)\left(\boldsymbol{\omega}^T\boldsymbol{\omega}\right)\delta q_4^2 \tag{43}$$

Hence, since $\beta > 0$ and $k > 0$, Equation (42) is indeed a Lyapunov function. The advantage of using $\operatorname{sgn}(\delta q_4)$ in the control law at all times (even before the sliding manifold is reached) now becomes clear. The step input in Equation (40) is a function of $\operatorname{sgn}(\delta q_4)$. Therefore, the response for δq_4 will approach $\operatorname{sgn}(\delta q_4)$ for any initial condition. This tends to drive the system to the desired location in the shortest distance. Furthermore, this inherently takes into account the rate errors as well. For example say that $\delta q_4(t_0)$ is positive, and that a high initial rate is given which tends to drive the system away from $\delta q_4 = 1$. The control law will automatically begin to null the rate. But, if the initial rate is large enough and the control dynamics are relatively slow, then δq_4 may become negative. Since $\operatorname{sgn}(\delta q_4)$ is used in the control system, then from Equation (40) the control law will subsequently drive the system towards $\delta q_4 = -1$. Therefore, using $\operatorname{sgn}(\delta q_4)$ at all times produces an optimal response for any type of initial condition error.

Robustness

In this section the robustness of the variable-structure controller with respect to modeling errors and external disturbances is addressed. This closely follows the approaches shown in Ref. [12]. We first define the following bounded modeling errors for the inertia matrix:

$$J = \hat{J} + \Delta J \tag{44a}$$

$$J^{-1} = \hat{J}^{-1} + \delta J \tag{44b}$$

where $\hat{J}$ is the nominal inertia matrix. Next, neglecting the gyroscopic term in Euler's equations, and adding an external disturbance input yields

$$\dot{\boldsymbol{\omega}} = J^{-1}\boldsymbol{u} + J^{-1}\boldsymbol{d} \tag{45}$$

where $\boldsymbol{d}$ denotes a bounded disturbance input. Under these conditions the time-derivative of the sliding manifold can be approximated by

$$\dot{s} \approx \delta J\,\hat{J}\left\{\frac{1}{2}k\operatorname{sgn}(\delta q_4)\left[\Xi^T(\boldsymbol{q})\Xi(\boldsymbol{q}_d)\boldsymbol{\omega}_d - \Xi^T(\boldsymbol{q}_d)\Xi(\boldsymbol{q})\boldsymbol{\omega}\right] + \dot{\boldsymbol{\omega}}_d - G\boldsymbol{\vartheta}\right\} - J^{-1}\hat{J}\,G\boldsymbol{\vartheta} + J^{-1}\boldsymbol{d} \tag{46}$$

where it is assumed that the higher-order perturbations in the inertia matrix are small. Also, we again assume that the thickness of the boundary layer ε and the gain G are sufficiently large to keep the time derivative of the associated Lyapunov function negative-definite with modeling errors and external disturbances. Then the dynamics of the sliding manifold reduce down to

$$\dot{s} \approx -\frac{1}{\varepsilon} J^{-1} \hat{J} G s + J^{-1} d \tag{47}$$

Therefore, if the time derivative of the sliding manifold is small after all transients have decayed, then s will satisfy the following inequality

$$\|s\| \leq \left\| \varepsilon \left(\hat{J} G \right)^{-1} \right\| \|d\|_{\max} \tag{48}$$

Equation (48) is valid using either external torques or reaction wheels in the control system.

ATTITUDE CONTROL OF MAP

In this section, the variable-structure controller is used to control the attitude of the Microwave Anisotropy Probe (MAP) spacecraft from quaternion observations and gyro measurements. The spacecraft is due to be launched around the year 2000. The objective of the MAP mission is to create a full-sky map of the cosmic microwave background and measure anisotropy with 0.3° angular resolution, in order to answer fundamental cosmological questions such as, inflationary versus non-inflationary "big bang" models, accurate determination of the Hubble constant, and the existence and nature of dark matter.

The ideal orbit for the MAP spacecraft is about the Earth-Sun L_2 Lagrange point, which is a Lissajous orbit with approximately a 180-day period. Because of its distance, 1.5 million km from Earth, this orbit affords great protection from the Earth's microwave emission, magnetic fields, and other disturbances, with the dominant disturbance torque being solar radiation pressure. It also provides for a very stable thermal environment and near 100% observing efficiency, since the Sun, Earth, and Moon are always behind the instrument's field of view. In this orbit MAP sees a Sun/Earth angle between 2 and 10 degrees. The instrument scans an annulus in the hemisphere away from the Sun, so the universe is scanned twice as the Earth revolves once around the sun.

The spacecraft orbit and attitude specifications are shown in Figure 1. To provide the scan pattern, the spacecraft spins about the z-axis at 0.464 rpm, and the z-axis cones about the Sun-line at 1 rev/hour. A 22.5°±0.25° angle between the z-axis and the Sun direction must be maintained to provide a constant power input, and to provide constant temperatures for alignment stability and science quality. The instrument pointing knowledge is 1.8 arcmin (1σ), which is not required for onboard or real-time implementation.

The spacecraft's attitude is defined by a 3-1-3 Euler angle rotation relative to a rotating, Sun-referenced frame. The three Euler angles are ϕ_d, θ_d, and ψ_d, and the desired states for observing mode are

$$\dot{\phi}_d = 1 \frac{\text{rev}}{\text{hr}} = 0.001745 \frac{\text{rad}}{\text{sec}} \tag{49a}$$

$$\theta_d = 22.5° = 0.3927 \text{rad} \tag{49b}$$

$$\dot{\psi}_d = 0.464 \text{ rpm} = 0.04859 \frac{\text{rad}}{\text{sec}} \tag{49c}$$

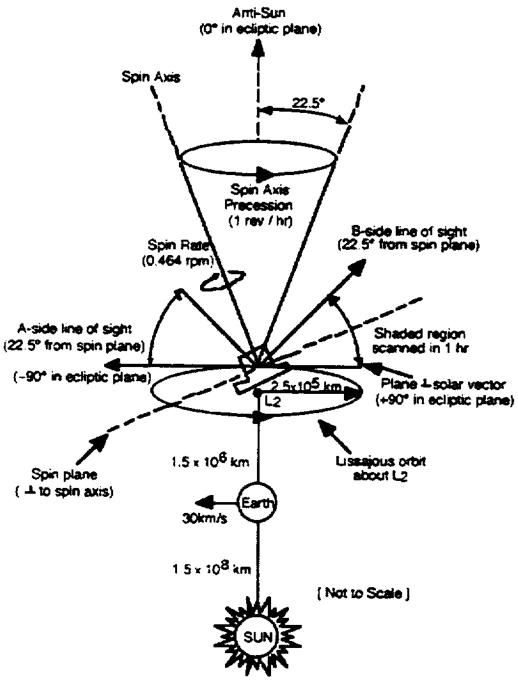


Fig. 1 MAP Spacecraft Specifications

The desired Euler angles for ϕ_d and ψ_d are determined by integrating the Euler rates. Also, $\dot{\theta}_d$ is set to zero. The commanded quaternion is determined using

$$q_{d_1} = \sin\left(\frac{\theta_d}{2}\right)\cos\left(\frac{\phi_d - \psi_d}{2}\right) \tag{50a}$$

$$q_{d_2} = \sin\left(\frac{\theta_d}{2}\right)\sin\left(\frac{\phi_d - \psi_d}{2}\right) \tag{50b}$$

$$q_{d_3} = \cos\left(\frac{\theta_d}{2}\right)\sin\left(\frac{\phi_d + \psi_d}{2}\right) \tag{50c}$$

$$q_{d_4} = \cos\left(\frac{\theta_d}{2}\right)\cos\left(\frac{\phi_d + \psi_d}{2}\right) \tag{50d}$$

The kinematic equation that transforms the commanded Euler rates to the commanded body rates is given by

$$\boldsymbol{\omega}_d = \begin{bmatrix} \sin\theta_d \sin\psi_d & \cos\psi_d & 0 \\ \sin\theta_d \cos\psi_d & -\sin\psi_d & 0 \\ \cos\theta_d & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi}_d \\ \dot{\theta}_d \\ \dot{\psi}_d \end{bmatrix} \tag{51}$$

A number of simulation studies have been performed in order to test the variable-structure controller performance. The first test case involves a slewing maneuver using the following initial condition for the actual quaternion:

$$\boldsymbol{q}(t_0) = \begin{bmatrix} 0 & 0 & \sin(\Phi/2) & \cos(\Phi/2) \end{bmatrix}^T \otimes \boldsymbol{q}_d(t_0) \tag{52}$$

where the angle of rotation error has been set to $\Phi = 60^\circ$. Also, the actual angular velocity has been set to the desired angular velocity. This case uses external torques to maneuver the spacecraft. The control system has been designed to bring the actual attitude to the desired attitude in less than 20 minutes. The gains used in the control law are: $k = 0.015$, $G = 0.0015\, I_{3\times3}$, and $\varepsilon = 0.01$. A plot of the roll, pitch, and yaw attitude errors are shown in Figure 2. Clearly, the control system achieves the desired performance. Plots of the angular velocity errors and control torques are shown in Figures 3 and 4, respectively. The sinusoidal components of the control torques in the X and Y axes are used to achieve the desired spin motion.

The next case shows how using $k\,\mathrm{sgn}[\delta q_4(t)]$ in the variable-structure controller achieves an optimal response (i.e., in the shortest distance). For this case, the initial attitude is given using Equation (52) with $\Phi = 60^\circ$ again. However, a large initial angular velocity error has been introduced, with $\boldsymbol{\omega}(t_0) = -5\,\boldsymbol{\omega}_d(t_0)$. This tends to drive the spacecraft in the opposite direction of the desired attitude. A plot of $\delta q_4(t)$ using $k\,\mathrm{sgn}[\delta q_4(t)]$ (solid line) in the controller and $\delta q_4(t)$ using just k (dotted line) in the controller, analogous to the approach in [8] and [9], is shown in Figure 5. Clearly, using $k\,\mathrm{sgn}[\delta q_4(t)]$ in the variable-structure controller achieves a faster response with less transients. This is also shown in the angular velocity responses, shown in Figures 6 and 7. Also, a plot comparing the magnitudes of the control inputs using $k\,\mathrm{sgn}[\delta q_4(t)]$ (solid line) and using k (dotted line) is shown in Figure 8. Using $k\,\mathrm{sgn}[\delta q_4(t)]$ results in a 22% reduction (root-mean-square) in the control effort to achieve the desired reference attitude and angular velocity.

To further illustrate the importance of using $k\,\mathrm{sgn}[\delta q_4(t)]$, a number of other simulations have been run for a rest-to-rest maneuver with the desired attitude set to the identity quaternion, and the desired angular velocity set to zero. The initial actual velocity has also been set to zero, and the initial quaternion has been computed using Equation (52). Test cases have been executed using $\Phi = 210^\circ$, 240°, 270°, 300°, and 330°. For all these test cases the spacecraft has been controlled using reaction wheels. A plot of the $(\delta\dot{q}_4, \delta q_4)$ phase plane is shown in Figure 9 (solid lines correspond to using $k\,\mathrm{sgn}[\delta q_4(t)]$ and dotted lines correspond to using k in the controller). Although not shown here, the response is the same for $0^\circ < \Phi < 180^\circ$. However, using $k\,\mathrm{sgn}[\delta q_4(t)]$ produces a response that is always optimal for any rotation error angle. Equation (28) has also been evaluated for each Φ used in this simulation. Table 1 summarizes the results for $k\,\mathrm{sgn}[\delta q_4(t)]$ and just k only. Clearly, by using $k\,\mathrm{sgn}[\delta q_4(t)]$ in the control law, better performance is achieved in the closed-loop system than using just k. This again shows that $k\,\mathrm{sgn}[\delta q_4(t)]$ is important in the variable-structure controller.

The final simulation shows how Equation (48) can be used to bound the steady-state errors. For this simulation the following external disturbance has been added:

$$\boldsymbol{d} = \begin{bmatrix} 0.005\sin(0.05t) \\ 0.003 \\ 0.005\cos(0.05t) \end{bmatrix} \tag{53}$$

A plot of the sliding manifold s and associated bound using Equation (48) is shown in Figure 10. This clearly shows the effectiveness of Equation (48) to accurately bound the sliding manifold errors.

Table 1 Cost Function Values for Various Φ

	Value of $\frac{1}{2}\int_{t_0}^{t_1} s^T s\ dt$	
Φ (deg)	k case	$k\,\mathrm{sgn}[\delta q_4(t)]$ case
210	1.0787	0.9313
240	0.8988	0.6543
270	0.6136	0.3566
300	0.3185	0.1300
330	0.1095	0.0194

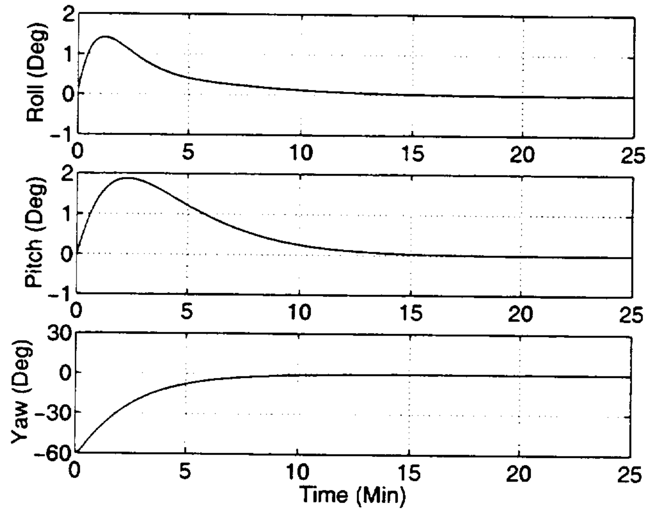


Fig. 2 Attitude Angle Errors

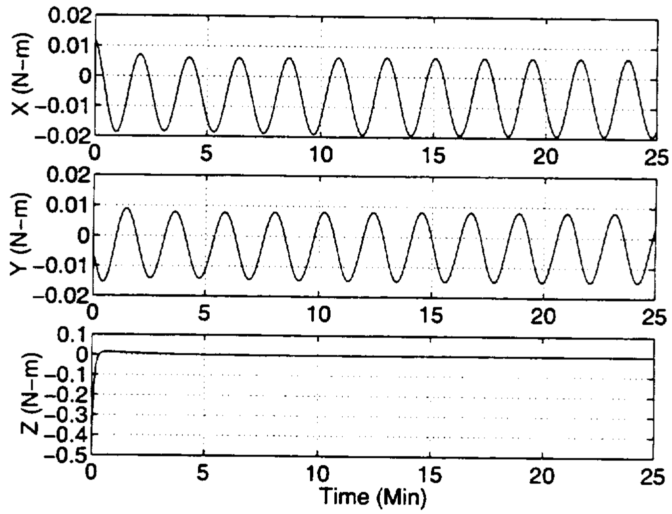


Fig. 4 Control Torque Inputs

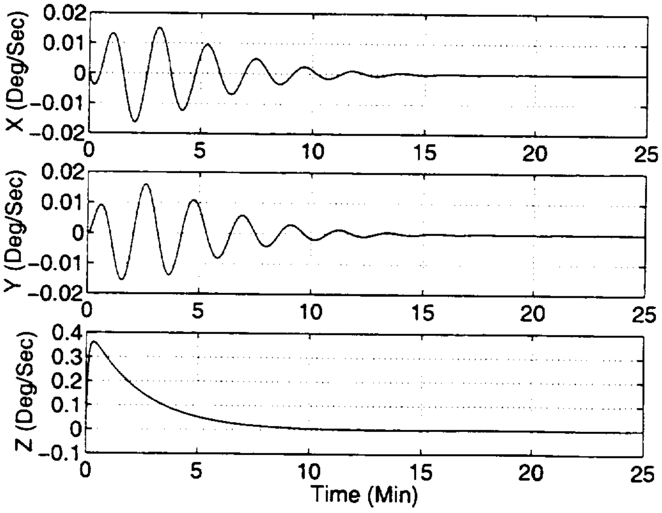


Fig. 3 Angular Velocity Errors

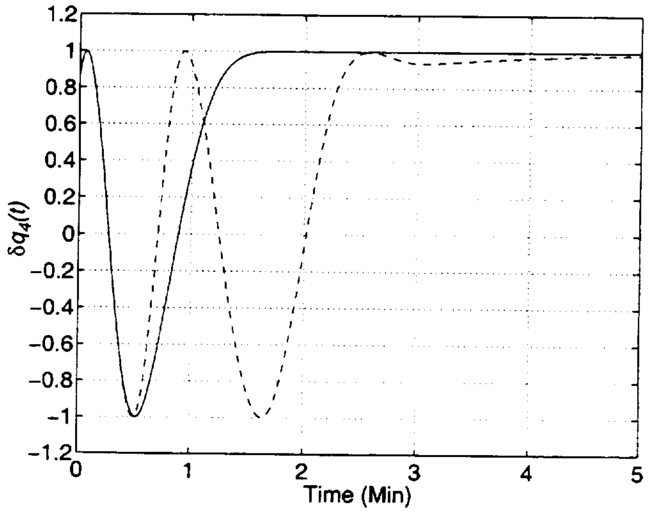


Fig. 5 Optimal (solid line) and Non-Optimal (dotted line) Error Quaternion Comparison

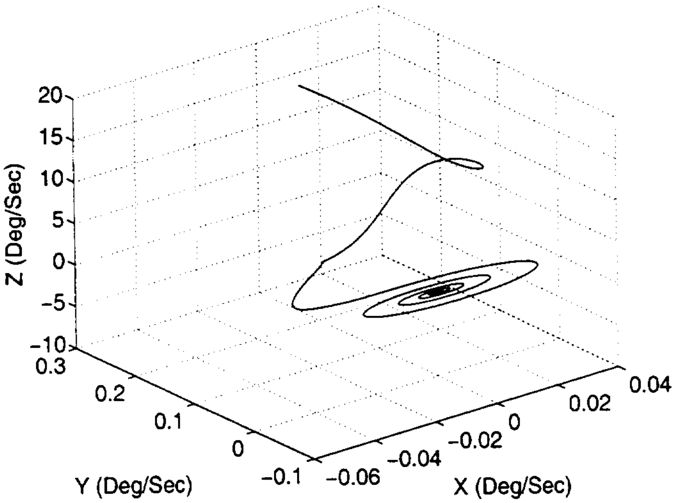


Fig. 6 Non-Optimal Angular Velocity Response

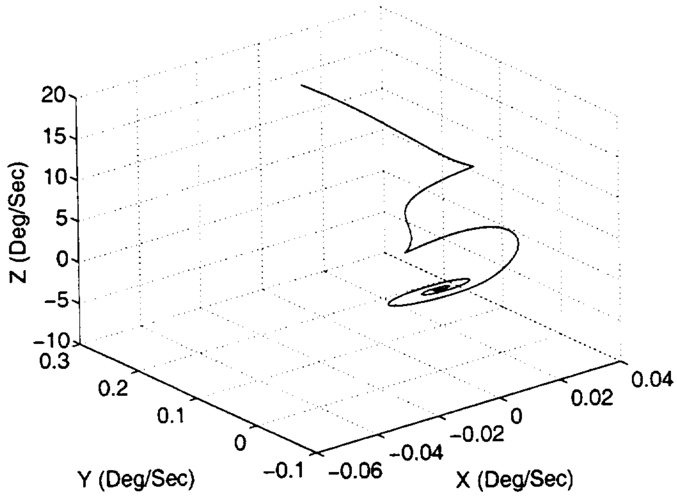


Fig. 7 Optimal Angular Velocity Response

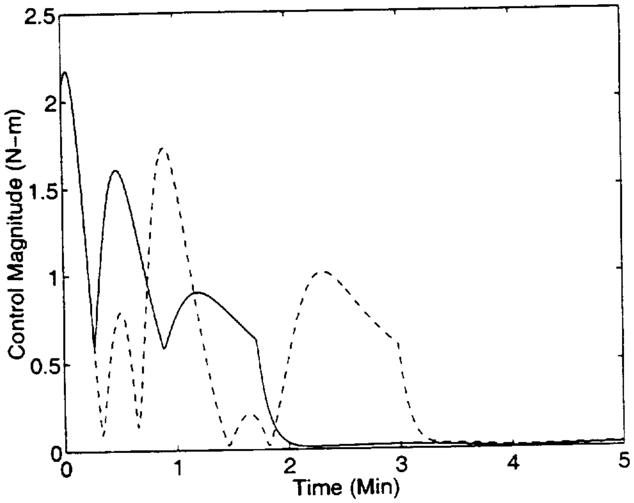


Fig. 8 Optimal (solid line) and Non-Optimal (dotted line) Control Input Magnitude Comparison

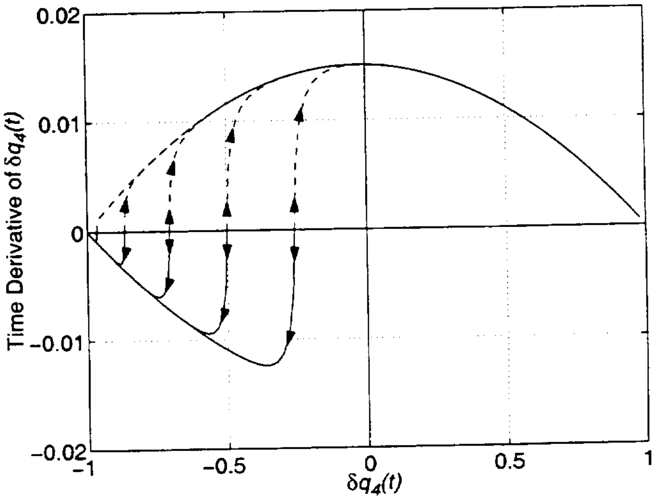


Fig. 9 Phase Portrait Plot

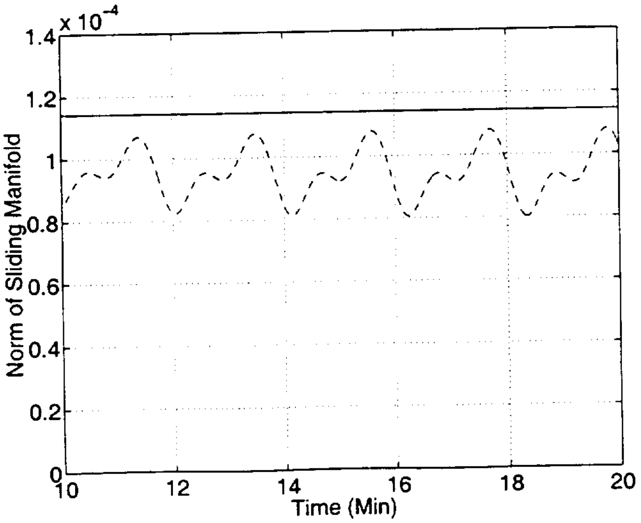


Fig. 10 Norm of the Sliding Manifold and Upper Bound

CONCLUSIONS

A new variable-structure controller for optimal spacecraft tracking maneuvers has been shown. The new controller was formulated for both external torque inputs and reaction wheel inputs. Global asymptotic stability was shown using a Lyapunov analysis. A simple term in the control law was used to produce a maneuver to the reference attitude trajectory in the shortest distance. The sliding motion was shown to be optimal in the sense of a quadratic loss function in the multiplicative error quaternions and angular velocities. A simulation study was performed which uses the new control law to stabilize the motion of the Microwave Anisotropy Probe spacecraft. Results indicated that the addition of the simple term in the control law always provides an optimal response, so that the reference attitude motion is achieved in the shortest possible distance.

REFERENCES

[1] Scrivener, S.L., and Thompson, R.C., "Survey of Time-Optimal Attitude Maneuvers," *Journal of Guidance, Control and Dynamics*, Vol. 17, No. 2, March-April 1994, pp. 225-233.

[2] Vadali, S.R., and Junkins, J.L., "Optimal Open-Loop and Stable Feedback Control of Rigid Spacecraft Maneuvers," *The Journal of the Astronautical Sciences*, Vol. 32, No. 2, April-June 1984, pp. 105-122.

[3] Wie, B., and Barba, P.M., "Quaternion Feedback for Spacecraft Large Angle Maneuvers," *Journal of Guidance*, Vol. 8, No. 3, May-June 1985, pp. 360-365.

[4] Slotine, J.J.E., and Li, W., *Applied Nonlinear Control*, Prentice Hall, NJ, 1991.

[5] Dwyer, T.A.W., and Sira-Ramirez, H., "Variable Structure Control of Spacecraft Reorientation Maneuvers," *Journal of Guidance, Control and Dynamics*, Vol. 11, No. 3, May-June 1988, pp. 262-270.

[6] Crassidis, J.L., and Markley, F.L., "Sliding Mode Control Using Modified Rodrigues Parameters," *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 6, Nov.-Dec. 1996, pp. 1381-1383.

[7] Vadali, S.R., "Variable-Structure Control of Spacecraft Large-Angle Maneuvers," *Journal of Guidance, Control and Dynamics*, Vol. 9, No. 2, March-April 1986, pp. 235-239.

[8] Robinett, R.D., and Parker, G.G., "Least Squares Sliding Mode Control Tracking of Spacecraft Large Angle Maneuvers," *Journal of the Astronautical Sciences*, Vol. 45, No. 4, Oct.-Dec. 1997, pp. 433-450.

[9] Lo, S.-C., and Chen, Y.-P., "Smooth Sliding-Mode Control for Spacecraft Attitude Tracking Maneuvers," *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 6, Nov.-Dec. 1995, pp. 1345-1349.

[10] Shuster, M.D., "A Survey of Attitude Representations," *The Journal of the Astronautical Sciences*, Vol. 41, No. 4, Oct.-Dec. 1993, pp. 439-517.

[11] Markley, F.L., "Equations of Motion," *Spacecraft Attitude Determination and Control*, Wertz, J.R., (editor), Kluwer Academic Publishers, Dordrecht, 1978, pp. 521-523.

[12] Kim, J., and Crassidis, J.L., "A Comparative Study of Sliding Mode Control and Time-Optimal Control," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Boston, MA, Aug. 1998, AIAA Paper #98-4473.

THERMAL SNAP OF SATELLITE SOLAR PANELS

John D. Johnston and Earl A. Thornton
Department of Mechanical and Aerospace Engineering, University of Virginia
Charlottesville, VA 22903

ABSTRACT

Thermal snap disturbances are known to affect the attitude dynamics of low Earth orbiting satellites during orbital eclipse transitions. Prominent examples of satellites experiencing the disturbances include the TOPEX, UARS and LANDSAT satellites. Thermal snap disturbances result from thermally-induced structural motions of flexible appendages such as deployable booms and solar arrays. Motions of flexible appendages lead to rigid body rotations of the entire satellite, since the total angular momentum of the system must be conserved. These potentially large attitude disturbances may violate mission pointing accuracy and jitter requirements. To fully understand thermal snap disturbances, it is necessary to develop a detailed understanding of the thermal-structural behavior of flexible satellite appendages. This paper describes recent studies of the thermal-structural performance of solar panels including an analysis of satellite attitude dynamics resulting from solar panel thermally-induced structural motions during orbital eclipse transitions and a laboratory investigation of the thermal-structural performance of a solar panel from NASA's Transition Region and Coronal Explorer (TRACE) satellite.

INTRODUCTION

The thermal-structural performance of deployable appendages can have a significant impact on the attitude dynamics and control of satellites. Nonuniform thermal loading can give rise to a cross-sectional temperature differences in appendages which, due to differential thermal expansion, result in structural deformations. Additionally, rapid changes in thermal loading initiated as a satellite exits or enters the Earth's shadow may excite dynamic structural motions.[1] Thermally-induced structural motions (TISM) of flexible satellite appendages may be classified as: thermal bending, thermal snap, thermally-induced vibrations, or thermal flutter. Thermal bending motions are quasi-static structural deformations resulting from slowly varying temperature differences. A quasi-static deformation consists of a succession of equilibrium displacements each corresponding to the temperature difference at a given instant in time. Since the temperature differences driving thermal bending motions develop slowly, appendage accelerations are very small. Thermal snap (or thermal elastic shock) motions involve rapid, non-oscillatory appendage deformations initiated during orbital eclipse transitions. A thermal snap response is similar to quasi-static TISM in that it consists of a succession of quasi-equilibrium displacements resulting from time-varying temperature differences in appendages. However, in the thermal snap case, the rapid rise and decay of the temperature differences result in brief acceleration transients in the appendage structural response at times corresponding to shadow crossings. Thermally-induced vibrations, which consist of a quasi-static deformation and superimposed oscillations, are a stable dynamic response and may involve bending, torsional, or combined bending and torsional structural motions. Thermally-induced vibrations typically occur when flexible appendages are subject to rapidly developing or decaying temperature differences such as when the satellite undergoes an orbital eclipse transition. The most severe type of thermally-induced structural motion is thermal flutter. Thermal flutter is an unstable thermally-induced vibrations response. The instability mechanism is coupling between incident heating and structural deformations. Thermal flutter may occur for both bending and torsional motions. Table 1 presents a compilation of cases where satellite performance was affected by TISM of solar arrays. The table presents a list of satellites known to have experienced solar array TISM, the satellite launch dates, and brief descriptions of the disturbances.[2]

A common solar array design is the rigid panel solar array, which consists of one or more solar panels each incorporating a honeycomb sandwich panel substrate onto which solar cells are mounted. Mathematical modeling of satellite attitude disturbances resulting from thermally-induced structural motions of rigid panel solar arrays has been explored by a number of authors. Jasper and Neste[3] studied the effect of rapid thermal bending on the dynamics of the LANDSAT and UARS satellites. Zimbelman[4,5] performed a detailed analysis of the disturbance affecting the TOPEX satellite and established an analytical model for predicting the disturbance torques resulting from thermally-induced deformations of rigid panel solar arrays. These studies utilized a quasi-static approach for formulating the solar array structural response and the corresponding disturbance torque. More recently, Johnston and Thornton[6-8] utilized a hybrid coordinate dynamical model to analyze the attitude dynamics of simple satellites with flexible booms and solar panels experiencing both quasi-static and dynamic thermally-induced structural motions initiated by orbital eclipse transitions.

This paper will summarize two recent studies carried out to investigate thermal snap of satellite solar panels. Full details of the studies are presented in Ref. 2. The first study utilizes an analytical formulation for the planar dynamics of a simple satellite with a solar panel undergoing thermally-induced structural motions to study the effects of TISM disturbances on satellite attitude dynamics. The second study involves an experimental investigation of the thermal-structural performance of a satellite solar panel. The investigation utilizes a solar panel assembly from the NASA TRACE satellite to characterize the thermal-structural performance of satellite solar array hardware under laboratory simulated orbital eclipse transition heating. The overall objective of this paper is to utilize the results of the analytical and experimental studies to provide a detailed explanation for satellite solar panel thermal snap disturbances.

THERMAL SNAP DISTURBANCES

The UARS satellite was predicted and observed to experience attitude disturbances during orbital eclipse transitions due to rapid thermal bending of its solar array. The disturbances are of sufficient magnitude to violate the stability requirements for some of the science instruments on the satellite. Figure 1(a) presents a schematic of the UARS satellite, which utilizes a single wing rigid panel solar array. Flight data from Ref. 9 demonstrating the attitude accelerations (roll, pitch, and yaw shown in arcsec/s^2) during a sunrise orbital eclipse transition is presented in Figure 1(b). The disturbance is manifested predominately in the roll (X) response, which is consistent with the geometry shown in Figure 1(a). There are also significant rotations about the pitch (Y) and yaw (Z) axes. The first peak in roll is approximately -1.8 arcsec/s^2 (-0.0005 deg/s^2), followed by another peak of +1.5 arcsec/s^2 (+0.0004 deg/s^2). The entire disturbance transient lasts in excess of 200 s. A thermal snap (or thermal elastic shock) disturbance history involves a short duration transient consisting of a impulse-like response followed by a decaying exponential of the opposite sign.[4] The observed UARS attitude acceleration history is consistent with a thermal snap disturbance resulting from rapid, non-oscillatory solar array deformations. In the following sections, the results of analytical and experimental studies of solar panel thermal-structural behavior will be used to explain the characteristics of the thermal snap disturbances observed in satellite flight data.

ANALYTICAL STUDIES

This section summarizes an analysis of the problem of thermally-induced structural motions of solar panels initiated during eclipse transitions and their effect on satellite attitude dynamics. The objectives of the studies are to: (1) predict the thermal response of solar panels to orbital eclipse transition heating, and (2) predict the planar dynamics of a simple satellite with a solar panel undergoing thermally-induced structural motions.

Thermal Analysis

The thermal analysis involves two steps: (1) calculation of the incident heating, and (2) calculation of the solar panel thermal response.

Orbital Eclipse Heating

The orbital eclipse heating analysis calculates the time-varying heating experienced by a satellite during orbital eclipse transitions. Consider the case of a sunrise eclipse transition. Initially, the satellite is in the region of total shadow called the umbra. The satellite then crosses through the region of partial shadow (penumbra), and finally emerges into full sunlight. The intensity of the incident solar heat flux experienced by a satellite in the penumbra is proportional to the area of the solar disk that is visible to the satellite. The time history of the incident heating must determined accurately to predict the solar panel thermal response. The heating analysis makes the following assumptions: (1) the satellite is in a circular orbit , (2) the orbit plane coincides with the plane of the Earth's orbit around the Sun (the ecliptic plane), and (3) variations in solar heating dominate the thermal response during eclipse transitions thus Earth-reflected (albedo) and Earth-emitted heat fluxes are neglected. The orbital eclipse heating analysis used in this study follows the analysis given by Baker[10] that calculates the actual solar heat flux during penumbral transitions based on the geometry of the Sun-Earth-satellite system. The intensity of the incident solar heating experienced by a satellite during passage through the penumbra is related to the area of the Sun's disk visible to the satellite. This area is determined by considering the angular sizes of the Earth/Sun disks and the angular separation between the centers of the Earth and Sun as viewed from the satellite.

Solar Panel Thermal Response

The thermal response of the solar panel subject to the calculated time-varying incident heating is now determined using finite element analysis. The results of the thermal analysis are required for use as input to solar panel thermal-structural models in the satellite dynamics analysis. Traditional thermal analyses for satellite structures are performed to predict surface temperature histories. For thermally-induced structural motions studies it is necessary to also calculate cross-sectional temperature difference histories and the first and second time derivatives of the temperature difference as functions of time.

The solar panel thermal model is presented in Figure 2. The analysis assumes one-dimensional conduction through the thickness of the solar panel (z direction) subject to radiation boundary conditions on the front and back surfaces. The boundary conditions consist of a uniform solar heat flux, S(t), directed normal to the front surface of the solar panel and thermal radiation from the front and back surfaces to a deep space temperature of 0 K. The solar heat flux, S(t), is calculated as described in the previous section. The analysis assumes that the solar panel is sun-tracking over the entire orbit, i.e. the solar heat flux is always incident normal to the front surface of the panel. Also, the absorbed heat flux is assumed to remain constant for small solar panel deformations, i.e. the thermal response is assumed to be independent of the structural response. The temperature of the solar panel, T(z,t), is assumed to be uniform in the x and y directions and to vary through the thickness of the panel in the z direction. The temperatures at the surfaces of the front and back facesheets are $T_{front}(t)$ and $T_{back}(t)$, respectively, and the temperature difference through the thickness of the solar panel is given by $\Delta T(t) = T_{front}(t)-T_{back}(t)$. The transient thermal response of the solar panel was solved using the commercially available finite element analysis program ABAQUS.[11] A one-dimensional finite element model of the solar panel was utilized in the analysis. The model used a total of 25 two-node rod elements through the thickness of the panel. Each of the different layers in the solar panel (facesheets, adhesive, and core) is represented in the model. Post-processing yields results for the temperature difference through-the-thickness of the panel, as well as the time derivatives of the temperature difference which are required for later solar panel structural response studies.

Satellite Dynamics Analysis

The objective of the satellite dynamics analysis is to investigate the effects of thermally-induced structural motions on the planar dynamics of a simple satellite consisting of a rigid hub and a flexible solar panel. The approach taken is based on a technique developed by Junkins and Kim[12] which employs a generalized form of Lagrange's equations to obtain the governing equations and boundary conditions for a wide class of hybrid coordinate multibody systems representative of satellites with flexible appendages. Using this approach, a new formulation for the problem of thermally-induced structural motions was presented by Johnston and Thornton in Refs. 6-8.

The problem considered is the planar motions of a simple satellite, Fig. 3, consisting of a rigid hub and a cantilevered flexible solar panel. The rigid hub has a mass moment of inertia, I_{hub}, and a radius, R_o. The flexible solar panel is modeled as a Bernoulli-Euler beam of length L, mass per unit length ρA, and bending stiffness EI. Two coordinate systems used in the analysis are also shown in Fig. 3. The I1-I2 axes are located in an inertial reference frame fixed with respect to motions of the satellite. The B1-B2 axes are located in a body fixed reference frame attached to the hub with the B1 axis coinciding with the neutral surface of the undeformed solar panel. The origins of both sets of axes coincide with the center of the hub. The attitude angle, θ(t), measures rigid body rotations of the hub about its center, and v(x,t) is the deformation of the flexible appendage relative to the B1 axis. Only planar motions of the system consisting of small rotations of the hub about its center and bending vibrations of the solar panel in the I1-I2 plane are considered. There are no external forces or moments acting on the system, so the total angular momentum of the system is conserved.

The governing equations for the dynamic response of the rigid hub-flexible solar panel system are obtained using a generalized form of Lagrange's equations for hybrid coordinate systems. The resulting governing equation for the rigid body rotations of the hub is given by

$$I_{sc}\ddot{\theta} + \int_0^L \rho A\left(R_o + x\right)\ddot{v}(x,t)\,dx = 0 \qquad (1)$$

where I_{sc} is the composite mass moment of inertia for the spacecraft. An expression for the thermally-induced disturbance torque, $T_{TID}(t)$, can be obtained from Eq. (1) by moving the terms corresponding to motions of the flexible solar panel to the right hand side of the equation,

$$T_{TID}(t) = -\rho A\int_0^L\left(R_o + x\right)\ddot{v}(x,t)dx \qquad (2)$$

Note that the thermally-induced disturbance torque is directly proportional to the second time-derivative of the solar panel accelerations. The governing equation for the solar panel deformations in the body fixed reference frame is given by

$$\rho A\left(R_o + x\right)\ddot{\theta} + \rho A\ddot{v} + c_{damp}\dot{v} + EIv^{IV} = 0 \qquad (3)$$

where $c_{damp} = 2\zeta\omega_n\rho A$ is the viscous damping constant, ω_n are the natural frequencies of the solar array, and ζ is the damping factor. The boundary conditions for the solar panel are

$$v(0,t) = 0 \qquad (4a)$$
$$v'(0,t) = 0 \qquad (4b)$$
$$EIv''(L,t) + M_T(t) = 0 \qquad (4c)$$
$$EIv'''(L,t) = 0 \qquad (4d)$$

where at x = 0 the solar panel is cantilevered, and at x = L the solar panel is free. Note that in Eq (4c) the usual boundary condition of zero moment at a free end is modified by the inclusion of the thermal bending moment, $M_T(t)$, which is given by

$$M_T(t) = EI\alpha_{cte}\Delta T(t)/h \qquad (5)$$

where α_{cte} is the coefficient of thermal expansion of the facesheets, ΔT(t) is the temperature difference through-the-thickness of the solar panel, and h is the separation distance between the facesheets. The time-dependent thermal moment acts as a forcing term in the solar panel equations of motion.

Quasi-static Response

A quasi-static response occurs for the case where inertia effects are negligible, i.e. when there are negligible vibrations. The quasi-static structural response consists of a succession of equilibrium displacements each corresponding to the temperature distribution at a given instant in time. The quasi-static displacements of the solar panel, $v_{qs}(x,t)$, are obtained by neglecting terms corresponding to inertia forces in Eq. (3). Solving Eq. (3) subject to the boundary conditions (4) yields the following solution for the quasi-static displacements

$$v_{qs}(x,t) = \frac{-\alpha_{cte}\left(1-v^2\right)\Delta T(t)}{2h}x^2 \qquad (6)$$

This result shows that for a positive ΔT, $T_{front} > T_{back}$, the solar panel will experience negative displacements, i.e. bend away from the direction of the incident heating. Note that the $(1-v^2)$ term accounts for the structure being a panel instead of a narrow width beam. The displacements vary spatially with the square of the distance x along the solar panel and vary in time with the through-the-thickness temperature difference. Differentiating Eq. (6) with respect to time yields the velocity and acceleration of the solar panel which are directly proportional to the first and second time derivatives of the through-the-thickness temperature difference. These derivatives are normally zero except for a short duration of time surrounding eclipse transitions when the temperature difference varies at a non-constant rate. During eclipse transitions, a disturbance torque arises due to the acceleration of the solar panel. The quasi-static thermally-induced disturbance torque, $T_{QSTID}(t)$, is obtained from Eqs. (2) and (6)

$$T_{QSTID}(t) = \frac{\rho A\alpha_{cte}\left(1-v^2\right)}{2h}\left(\frac{R_oL^3}{3}+\frac{L^4}{4}\right)\ddot{\Delta T}(t) \qquad (7)$$

Equation (7) shows that the quasi-static thermally-induced disturbance torque is directly proportional to the second time derivative of the temperature difference through-the-thickness of the solar panel. The actual disturbance torque includes additional terms due to solar panel vibrations.[2] The following section describes the solution for the dynamic response of the system.

Dynamic Response

An approximate form of the equations of motion stated in Eqs. (1) and (3) is obtained using the quasi-static solution and modal expansion. The assumed form of the solution is

$$v(x,t) = v_{qs}(x,t) + \Sigma_{n=1}^{N} q_n(t)\phi_n(x) \qquad (8)$$

where $v_{qs}(x,t)$ is the quasi-static response, $q_n(t)$ are the nth generalized modal coordinates, $\phi_n(x)$ are the nth shape functions, and N is the number of modes. The shape functions are chosen to be the eigenfunctions from the free vibration response of a fixed-base cantilevered beam. After the assumed form of the displacements is substituted into the governing equations, the result is a system of N+1 coupled linear ordinary differential equations for the discrete coordinate, $\theta(t)$, and the generalized modal coordinates, $q_n(t)$. These equations can be written in matrix form as

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{F(t)\} \qquad (9)$$

where $\{x\}^T = \{\theta \; q_1 \; q_2 \; \cdots q_N\}$ are the generalized coordinates for the system, and [M], [C], and [K] are N+1 by N+1 constant coefficient matrices. The forcing vector {F(t)} contains terms involving time derivatives of the quasi-static solar panel displacements that act as forcing functions on the discrete and generalized modal coordinates. Recall that the time derivatives of the quasi-static displacements are functions of the time derivatives of the temperature difference through-the-thickness of the solar panel.

Thus, both the temperature difference and its time derivatives are key parameters in the study of thermally-induced structural motions. Numerical integration of the governing equations leads to solutions for the thermally-induced dynamics of the satellite.

Numerical Example

The following numerical example illustrates the effect of thermal snap type thermally-induced structural motions of a solar panel on the planar attitude dynamics of a hub-solar panel system. First, results for the thermal response of the solar panel are given. Then, the resulting dynamic motions of the hub-solar panel system are presented.

Thermal Response

Finite element analysis was used to predict the thermal response of a representative solar panel subject to orbital eclipse transition heating. The analysis considers the case of a 600 km circular orbit whose orbital plane lies in the ecliptic. This altitude is representative of typical low Earth orbital satellite altitudes, such as that used by the UARS satellite. The parameters used in the study, Table 2, are representative of typical satellite solar panels. Additionally, the front surface of the solar panel has an absorptivity $\alpha = 0.79$ and an emissivity $\varepsilon = 0.81$, while the back surface has an emissivity $\varepsilon = 0.86$.

The sunrise orbital eclipse transition thermal response for the solar panel is presented in Fig. 4. .At t = 0 the solar panel is in total shadow, and at t = 10 s it enters the penumbra and begins the transition to full sunlight. The incident solar heat flux history is given in Fig. 4(a). The transition time between total shadow and full sunlight is approximately 8.5 s. Figures 4(b)-(c) present plots of the surface temperatures and temperature difference as functions of time. Subsequent to entering the penumbra, the surface temperatures begin a gradual rise, while the temperature difference rapidly develops to reach its steady-state value of 11 K in approximately 60 s. As noted earlier, the time derivatives of the temperature difference are needed for thermally-induced structural motions (TISM) studies. In the thermally-induced satellite dynamics analysis presented it was shown that the temperature difference derivatives act as forcing terms in the governing equations for the dynamic response of the system. Plots of the first and second time derivatives of the temperature difference are given in Figs. 4(d)-(e). The derivatives of the temperature difference were calculated using central difference approximations. The rate of change in the temperature difference, Fig. 4(d), reaches a peak value of +0.4 K/s at t = 18s. The second time derivative of the temperature difference, $d^2(\Delta T)/dt^2$, is shown in Fig. 4(e). The first peak in the transient occurs at t = 13 s and has a value of +0.13 K/s^2. The transient has a value of zero at the t = 18 s corresponding to the peak rate of change in the temperature difference, then reaches a second peak of -0.09 K/s^2 at t = 19 s. Note that the $d^2(\Delta T)/dt^2$ transient consists of an impulse-like response followed by a decaying exponential of the opposite sign, which is the same form as the thermal snap time history described earlier.

Satellite Dynamics Response

The equations of motion, Eq. (9), were numerically integrated using the central differences method to determine a solution for the dynamic response of the hub-solar panel system. The simulation utilized N = 5 flexible modes, with a timestep based on the highest mode included. The rigid hub has a mass of 5000 kg and a radius of 1.0 m, while the solar panel has a length of 9 m and a width of 3 m. Additionally, the solar panel has a mass per unit length, ρA, of 7.4 kg/m and a bending stiffness, EI, of 2.0E+04 N-m^2.

The solar panel thermal-structural response is given in Fig. 5. The temperature difference through the thickness of the solar panel, Fig. 5(a) has a steady state value of 11 K with a rise time of approximately 60 s. Figure 5(b) presents a plot of the tip displacement in the body fixed reference frame as a function of time. The tip displacement, v(L,t), has a steady state value of -0.4 m with negligible superimposed oscillations. The maximum peak to peak amplitude of the oscillations (solar panel jitter), δv, is less than 5.0E-05 m and is too small to be seen in the figure. The solar panel motions may be classified as a quasi-static response. The solar panel exhibits short duration, approximately 100 s, velocity and acceleration transients which are initiated as the satellite enters the penumbra. Note that the duration of the transients corresponds closely to the rise time for the temperature difference response. The tip velocity, Fig. 5(c), has a peak value of -0.026 m/s which occurs at t = 18 s corresponding to the time at which the temperature difference reaches its maximum rate of change (See Figure 4). The acceleration at the free end of the solar panel, Fig. 5(d), exhibits a characteristic thermal snap history with the addition of a small superimposed high frequency (0.5 Hz) component corresponding to the fundamental frequency of the coupled rigid hub-solar panel system. The values of the first and second peak of the acceleration transient are -0.0061 m/s^2 and +0.0032 m/s^2, respectively. It is interesting to note that even though vibrations are negligible in the displacement response, their presence is still manifested in the accelerations.

The rigid hub response is presented in Fig. 6. The solar panel motions result in a internal disturbance torque that acts to change to orientation of the hub. The thermally-induced disturbance torque, $T_{TID}(t)$, shown in Fig. 6(a) displays a characteristic thermal snap history consisting of an impulsive response followed by a decaying exponential of the opposite sign. The disturbance torque has overall peak values of +1.1 / -0.6 N-m with small amplitude damped oscillations at a frequency of 0.5 Hz due to low level appendage dynamics. Due to the presence of the oscillatory terms, the dynamic model, Eq. (2), predicts values for the first and second peaks in the torque which are respectively 30% and 15% greater than the quasi-static model from Eq. (6). Thus, it is important to include both quasi-static and dynamic terms when calculating thermally-induced disturbance torques. The attitude response of the rigid hub is presented in Figs. 6(b)-(d). The attitude angle, θ, response of the hub, Fig. 6(b), consists of a slowly developing rotation in the direction opposite of the solar panel motions. The attitude angle has a steady state value of 0.01 rad with negligible jitter (attitude angle jitter = $\delta\theta$ < 1E-05 rad). The attitude response displays rate and acceleration transients similar to the solar panel response, but with the signs reversed since the hub motions are always opposite those of the solar panel. The attitude rate, Fig. 6(c), has a peak value of 9.1E-04 rad/s that occurs at the same time as the peak in the solar panel velocities. The attitude acceleration, Fig. 6(d), exhibits the characteristic thermal snap history with peak values of +2.2E-04 and -1.2E-04 rad/s^2.

Discussion

Recall from the satellite dynamics analysis that for a quasi-static response the solar panel accelerations and the corresponding disturbance torque acting on the rigid hub are directly proportional to the second time derivative of the temperature difference, $d^2(\Delta T)/dt^2$. Thus, thermal snap disturbances are characterized by the form of the $d^2(\Delta T)/dt^2$ transients that consist of an impulse-like response followed by a decaying exponential of the opposite sign. In contrast to the thermal snap disturbances illustrated here, a thermally-induced vibrations disturbance transient exhibits oscillatory behavior and is larger in magnitude and longer in duration due to the sustained solar panel vibrations. Comparing the thermally-induced disturbance torques for these two cases shows that thermal snap disturbances are dominated by accelerations resulting from $d^2(\Delta T)/dt^2$ transients, while thermally-induced vibrations disturbances are dominated by accelerations resulting from solar panel vibrations. While considerable attention has been focused on the analysis of satellite disturbance data and analytical modeling of these phenomena, relatively few experimental studies have examined the thermal-structural behavior of satellite appendages. In the following section, the details of an experimental investigation performed to study solar panel thermal-structural behavior in the laboratory are presented.

EXPERIMENTAL STUDY

A complete understanding of the thermal-structural performance of solar panels cannot be obtained without experiments to validate analytical predictions, yet such investigations of this phenomenon are rare. For the experimental study the following objectives have been undertaken: (1) to design and fabricate a test fixture to study the thermal-structural performance of solar panels subject to rapid changes in thermal loading, (2) to provide data for validation of analytical studies, and (3) to investigate the behavior of solar panel flight hardware to gain further insight into thermal-structural phenomena. The following sections describe the laboratory test setup for the solar panel experiments and representative test results demonstrating key aspects of the thermal-structural behavior.

Test Set-up

The laboratory experiments are designed to subject a solar panel test article to radiant heating on one face and to measure the resulting thermal and structural responses. A photograph of the test setup for the TRACE solar panel experiments is presented in Fig. 7. The test setup consists of the following elements: a backstop support structure to which the solar panel is mounted, an array of infrared lamps which act as the radiant heat source, the TRACE solar panel test article, and instrumentation for recording the solar panel response. The backstop is constructed of structural steel I-beams mounted directly to the wall of the laboratory and acts as a rigid support structure. The solar panel is attached via a mounting plate that serves as an interface between the solar panel deployment hinges and the backstop. Radiant heating is provided by an array of quartz tube / tungsten filament infrared lamps capable of heating an area of approximately 1.3 m (52 in) by 0.8 m (32 in). The lamp array is located at a fixed stand-off distance of 400 mm from the front face of the solar panel and is mounted such that the heat flux is directed normal to the surface of the panel. The overall size of the TRACE solar panel is 0.96 m (37.75 in) by 0.51 m (20 in). The panel utilizes a metallic sandwich panel construction consisting of 2.54E-04 m (0.01 in) thick AL 6061-T6 facesheets adhesively bonded to a 0.0095 m (0.375 in) thick AL 5056-H39 honeycomb core. The surfaces of the solar panel where painted with high temperature paint to provide known surface properties. The test setup uses the same design deployment hinges to attach the solar panel to the mounting plate as are used to attach the actual solar panels to the satellite bus.

The experiments characterize both the transient and steady-state response of the test article in terms of front (heated) and back (unheated) surface temperatures, through-the-thickness temperature differences, and displacements at the free end of the panel. Thermocouples are used to measure the temperature at 17 points on the heated surface and 5 points on the unheated surface, and provide a measurement of the through-the-thickness temperature difference at 5 points. Displacements at the free end of the solar panel are measured using laser displacement sensors. These sensor provide non-contact measurement of the displacements with a resolution of approximately 0.01 mm. Three sensors are used to measure displacements at a fixed distance of 0.91 m (35.875 in) from the fixed end of the panel at three heights corresponding to the top edge, centerline, and bottom edge of the panel. During post-processing of the test results, the velocity and acceleration of the solar panel are calculated from the displacement data using finite difference approximations. The procedure for a typical test is as follows: (1) a test run begins when the data acquisition system is activated and 20 s of data are obtained with the test article at room temperature, (2) the infrared lamp array is turned-on simulating the rapid heating which occurs as a satellite exits from the Earth's shadow and remains at a constant power level for a duration of 2000s, and (3) the infrared lamp array is turned off simulating the rapid cool-down which occurs as a satellite crosses into orbital eclipse.

Representative Test Results

Results are now presented which demonstrate the thermal-structural response of the TRACE solar panel. Figure 8 presents representative results for the through-the-thickness temperature difference, displacements, velocity, and acceleration of the solar panel for the first 75 s of a typical test run. The temperature difference at the geometric center of the panel, Fig 8(a), rises rapidly following power-up of the infrared lamp array at t = 20s, and reaches a steady-state value of 9 K approximately 200s later. The temperature difference through the thickness of the panel results in a time-varying thermal moment which causes the panel to bend away from the direction of the applied heating. The displacements at the free end of the solar panel, Fig. 8(b), reach steady state values of – 5 mm in approximately 200s. The displacements develop on the same time scale as the temperature difference, and the response is a quasi-static thermally-induced structural motion. Recall that even though the displacements are classified as quasi-static, rapid structural motions during heat-up result briefly in acceleration transients that impart disturbance torques on satellites. Figures 8(b)-(c) present plots of the calculated velocity and acceleration at the free end of the solar panel. During heat-up, there is a negative velocity transient, Fig. 8(c), as the solar panel moves away from the heat source. The transient consists of a rapid change from the initial zero velocity state to a maximum value of –0.32 mm/s. The acceleration transient, Fig. 8(d), consists of two parts. The first part involves an impulse-like response followed by a zero crossing at the time corresponding to the maximum velocity. The second part of the transient consists of another peak which has the opposite sign of the first peak followed by a decaying exponential response to zero. The first and second peak values of the heat-up acceleration transient are –0.15 mm/s^2 and 0.03 mm/s^2. Note that the acceleration transient observed in the solar panel laboratory experiments has the same form as the solar panel acceleration transient predicted by the analytical solution. Thus, the acceleration transients observed in the experiments are representative of the thermal snap disturbances known to affect satellites with rigid panel solar arrays.

CLOSING REMARKS

This paper has presented analytical and experimental investigations of solar panel thermal snap disturbances. The following insights were gained from the thermally-induced satellite dynamics analysis and the solar panel thermal-structural experiments:

(1) The critical aspects of the thermal response of rigid panel solar arrays (solar panels) are the through-the-thickness temperature difference and its time derivatives. Sudden changes in solar panel heating, such as during orbital eclipse transitions or the power-up of the infrared lamp array in the laboratory tests, result in the rapid development or decay of temperature differences and the cause short duration transients in the histories of the first and second time derivatives of the temperature difference.

(2) The solar panel structural response is driven by the time-varying through-the-thickness temperature difference and may consist of either quasi-static or thermally-induced vibrations motions. Solar panels undergoing quasi-static deformations will experience brief thermal snap acceleration transients during heat-up and cool-down, while solar panels experiencing thermally-induced vibrations exhibit acceleration histories involving decaying oscillations.

(3) Rapid structural motions during heat-up and cool-down result in acceleration transients that impart a disturbance torque on the main body of a satellite. The disturbance torque is directly proportional to the solar panel accelerations, thus time history of the torque has the same form as that of the accelerations. Thermally-induced disturbance torques from thermal snap disturbances are dominated by solar panel accelerations resulting from $d^2(\Delta T)/dt^2$ transients, while torques from thermally-induced vibrations disturbances are dominated by solar panel accelerations resulting from vibrations of the system.

Finally, comparison of the attitude disturbances observed in flight data from the UARS satellite and the disturbance transients predicted by the satellite dynamics analysis and observed in the laboratory experiments reveals the same characteristic thermal snap time history in each case. While this research has provided significant insight into solar panel thermal snap, ultimately the phenomena should be the focus of a space-based experimental study. This research is necessary to overcome the limitations of ground-based testing of space structures and to provide in-situ measurements of thermally-induced structural motions for validation of analytical models for the behavior of satellite appendages in the space environment.

REFERENCES

1. Thornton, E.A., *Thermal Structures for Aerospace Applications*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Inc., Washington, D.C., 1996.
2. Johnston, J.D., "Thermally-Induced Structural Motions of Satellite Solar Arrays," Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, University of Virginia, Charlottesville, VA, May 1999.
3. Jasper, P.E. and Neste, S., "UARS Solar Array Snap," PIR No. U-1K21-UARS-481, General Electric Space Division, Philadelphia, PA, July 9, 1986.
4. Zimbelman, D.F., "Thermal Shock and its Effect on Spacecraft Attitude Control," Ph.D. Dissertation, University of Colorado, Boulder, CO, 1990.
5. Zimbelman, D.F., Dennehy, C.J., Welch, R.V., and Born, G.H., "A Technique for Optimal Temperature Estimation for modeling Sunrise/Sunset Thermal Snap Disturbance," *Journal of Spacecraft and Rockets*, Vol. 28, No. 4, 1991, pp. 448-456.

6. Johnston, J.D. and Thornton, E.A., "Thermally-Induced Attitude Dynamics of a Spacecraft with a Flexible Boom," *AAS/AIAA Spaceflight Mechanics Symposium*, Huntsville, AL, Feb. 10-12, 1997.
7. Johnston, J.D. and Thornton, E.A., "Thermally-Induced Structural Disturbances of Rigid Panel Solar Arrays," Proceedings of the 1997 Flight Mechanics Symposium, NASA Goddard Space Flight Center, May 19-22, 1997, NASA CP 3345, pp. 33-47.
8. Johnston, J.D. and Thornton, E.A., "Thermally-Induced Attitude Dynamics of a Spacecraft with a Flexible Appendage," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, July-August, 1998, pp. 581-587.
9. Lambertson, M., Underwood, S., Woodruff, C., and Garber, A., "Upper Atmosphere Research Satellite Attitude Disturbances During Shadow Entry and Exit," Paper No. AAS 93-319, 1993.
10. Baker, R.M.L., *Astrodynamics: Applications and Advanced Topics*, Academic Press Incorporated, New York, 1967.
11. *ABAQUS User's Manual*, Version 5.7, Hibbett, Karlsson, & Sorenson, Inc., 1997.
12. Junkins, J.L. and Kim, Y., *Introduction to Dynamics and Control of Flexible Structures*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Inc., Washington, D.C., 1993.

Table 1: Compilation of solar array thermally-induced structural motions disturbances.

Satellite	Date	Description
ADEOS	1997	TISM of flexible blanket solar array, Failure of solar array due to thermal expansion and contraction solar cell blanket beyond limits of solar array tension control mechanism
Space Flyer Unit (SFU)	1996	TISM of flexible blanket solar array during orbital eclipse transitions
Mir space station	1995	TISM of a solar array wing on the Kvant-II module observed during orbital eclipse transitions
INSAT-II	1994	Thermal bending of solar array and solar sail mast
TOPEX	1992	"Thermal snap" disturbances during eclipse transitions attributed to rapid thermal bending of large single wing solar array
Upper Atmosphere Research Satellite (UARS)	1991	"Thermal snap" disturbances during eclipse transitions attributed to rapid thermal bending of large single wing solar array
Hubble Space Telescope (HST)	1990	"Jitter" phenomenon attributed to thermally-induced vibrations of FRUSA type solar arrays
LANDSAT 4 / 5	1980s	"Thermal snap" experienced during orbital eclipse transitions due thermal bending of large single wing solar array
Communications Technology Satellite (CTS)	1978	Three-axis stabilized satellite experienced "Thermal elastic shock" during eclipse transitions

Table 2: Solar panel parameters for thermal response numerical examples.

Parameter	Facesheet	Adhesive	Honeycomb Core
Material	aluminum 6061-T6	FM123 (epoxy based)	aluminum 5056
t (m)	2.54E-04	1.27E-04	0.0254
k (W/m-K)	168	0.4	1.2
ρ (kg/m^3)	2800	1150	30.0
c (J/kg-K)	960	750	920

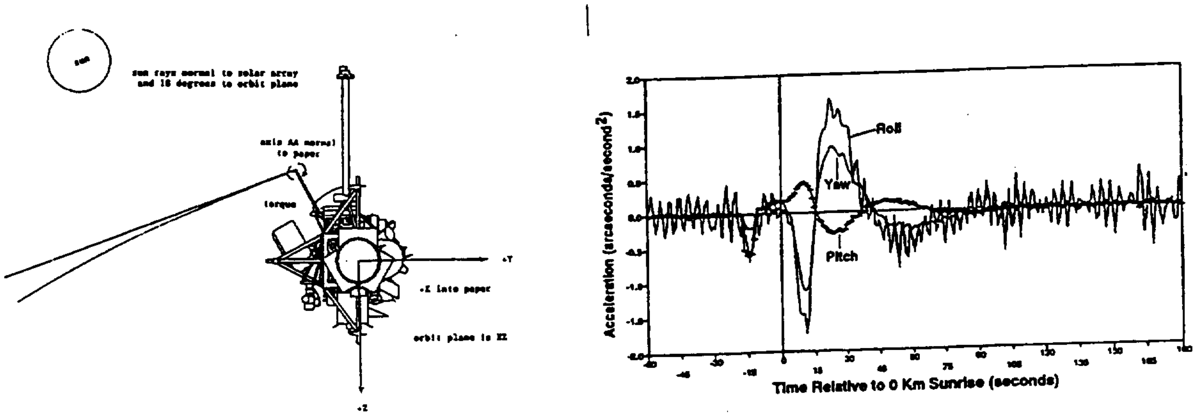


Figure 1: UARS satellite thermal snap disturbance: (a) Schematic of UARS satellite and solar array, (b) UARS attitude acceleration response for sunrise orbital eclipse transition. (Reference 9)

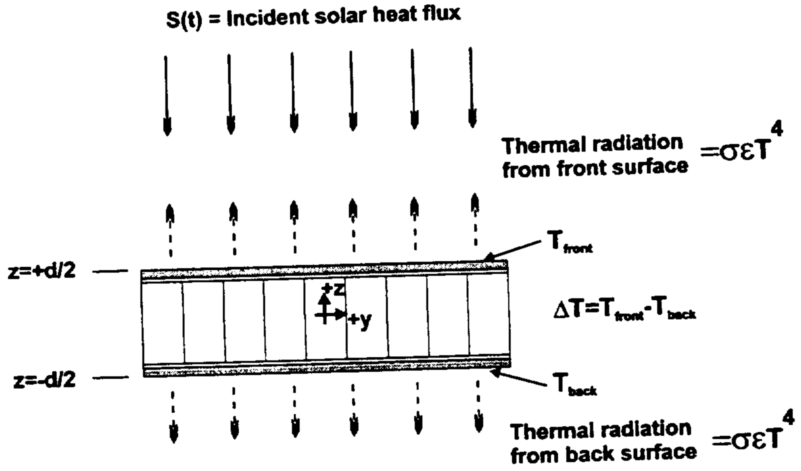

S(t) = Incident solar heat flux
Thermal radiation from front surface $=\sigma\varepsilon T^4$
T_{front}
z=+d/2
+z
+y
$\Delta T=T_{front}-T_{back}$
z=-d/2
T_{back}
Thermal radiation from back surface $=\sigma\varepsilon T^4$

Figure 2: Solar panel thermal model.

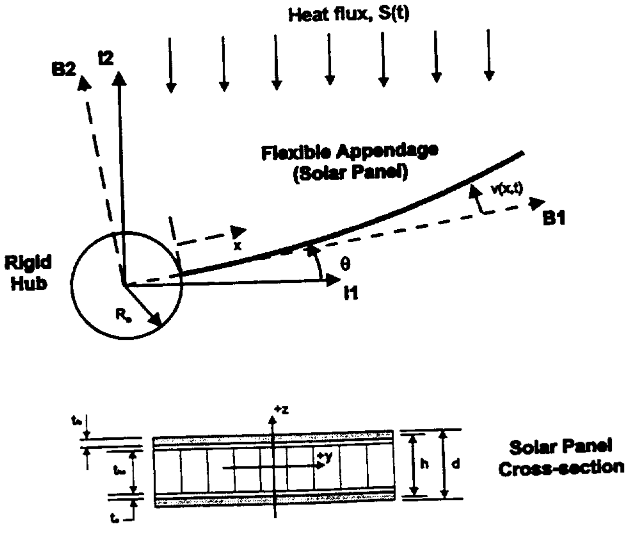


Figure 3: Satellite mathematical model and coordinate systems.

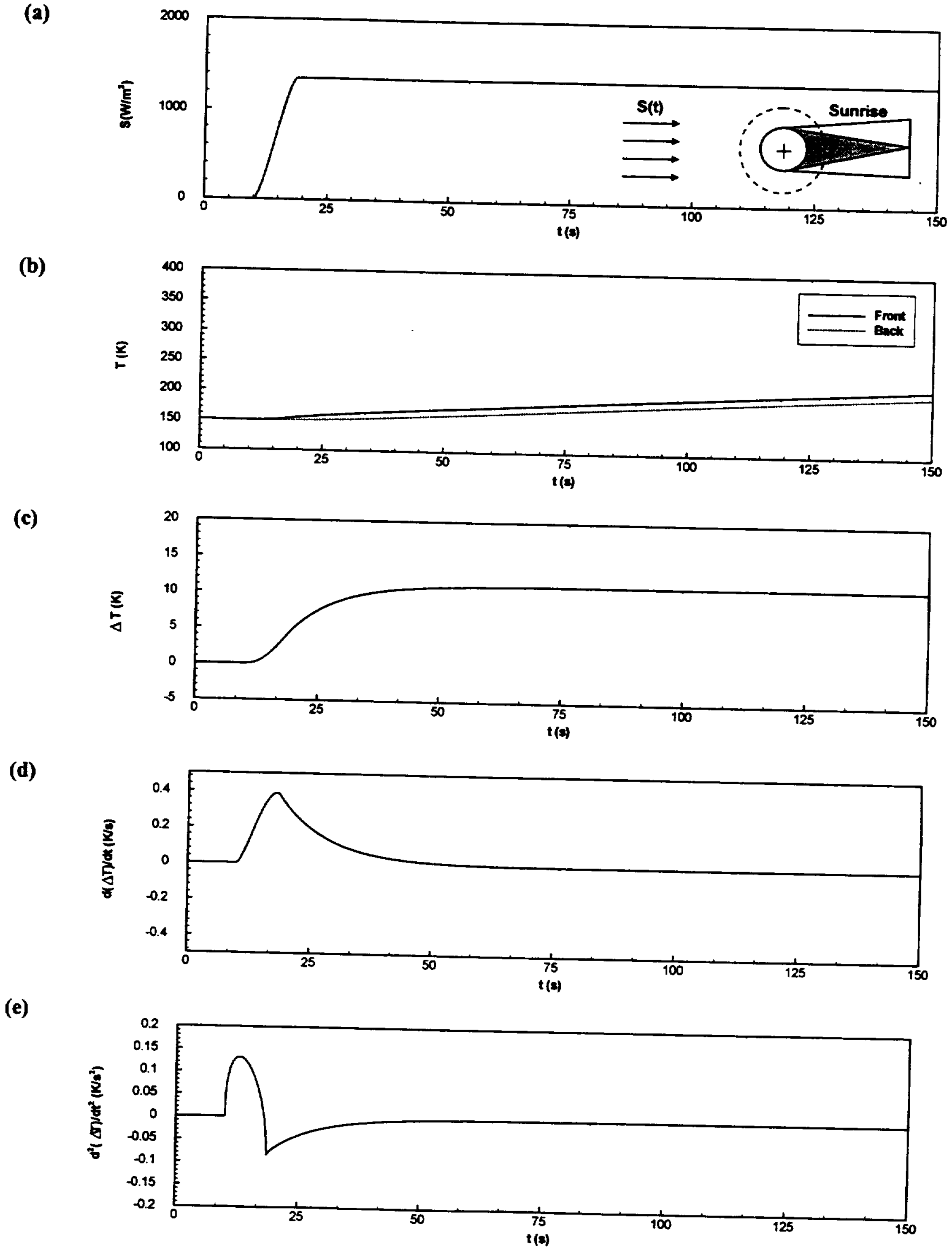


Figure 4: Solar panel thermal response at orbital sunrise (600 km circular orbit): (a) solar heating history, (b) surface temperatures, (c) through-the-thickness temperature difference, (d) first time derivative of temperature difference, and (e) second time derivative of temperature difference.

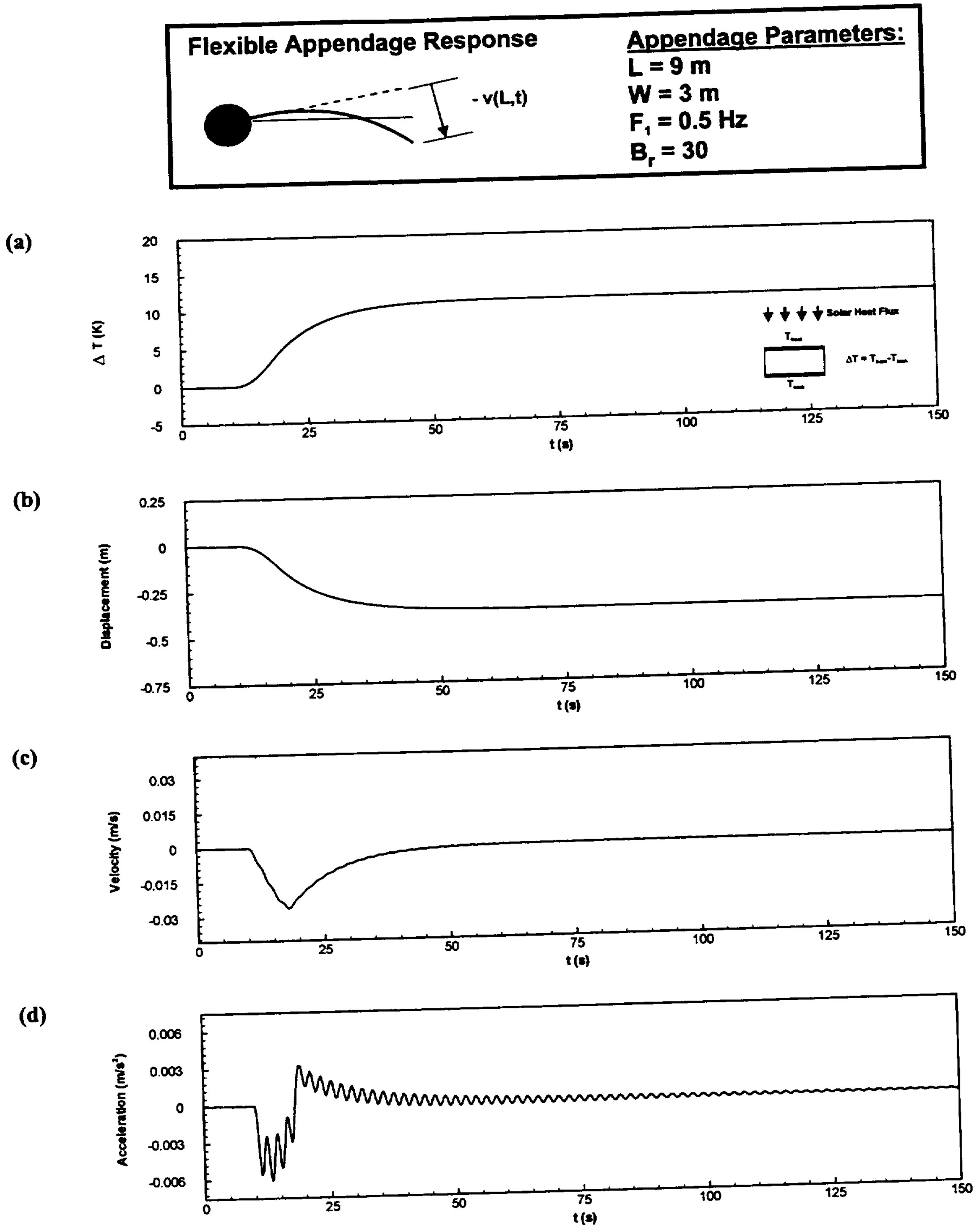


Figure 5: Solar panel thermal-structural response for *sunrise* eclipse transition (600 km circular orbit): (a) Through-the-thickness temperature difference vs. time, (b) tip displacement vs. time, (c) tip velocity vs. time, and (d) tip acceleration vs. time.

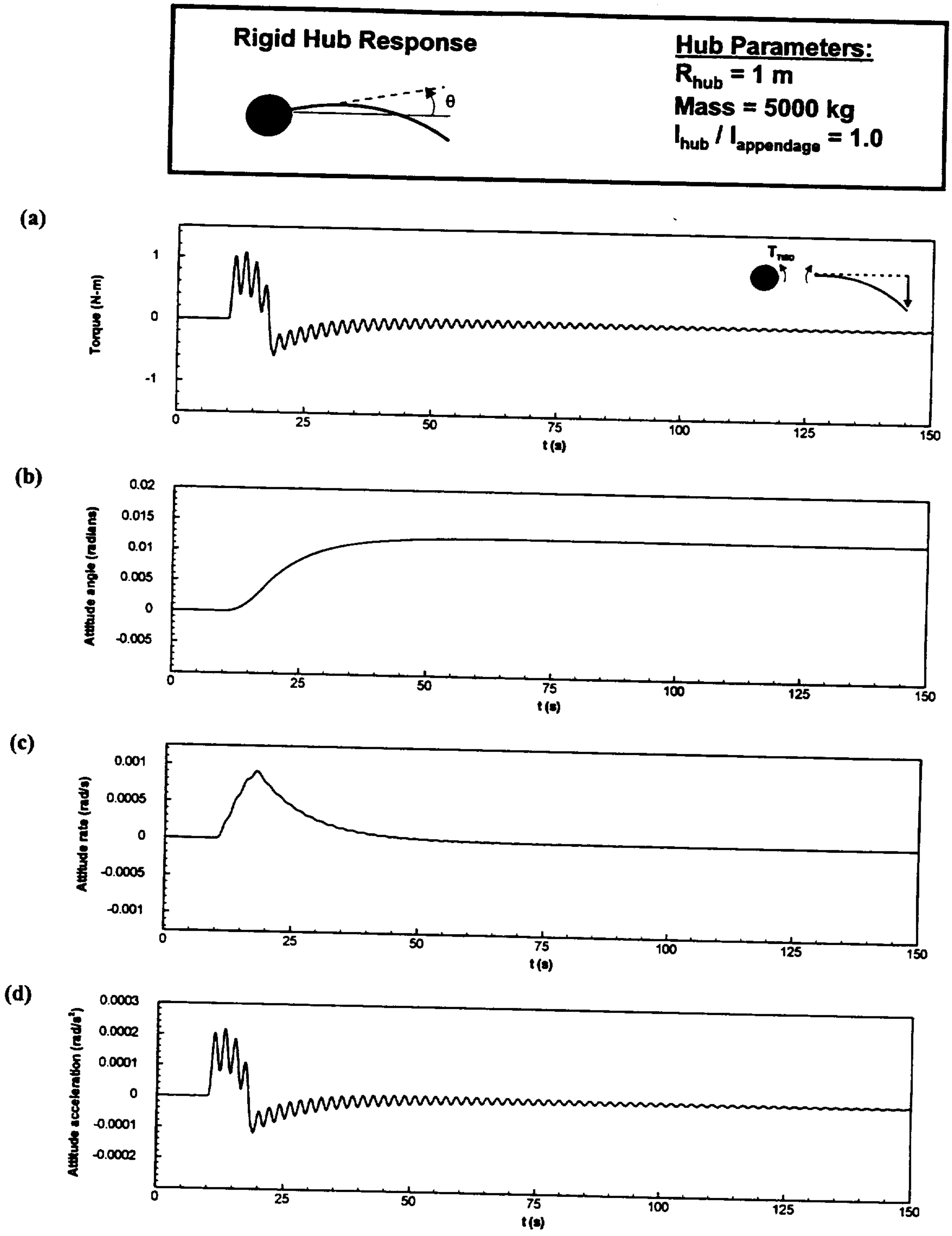


Figure 6: Rigid hub dynamic response for *sunrise* eclipse transition (600 km circular orbit): (a) Disturbance torque vs. time, (b) attitude angle vs. time, (c) attitude rate vs. time, and (d) attitude acceleration vs. time.

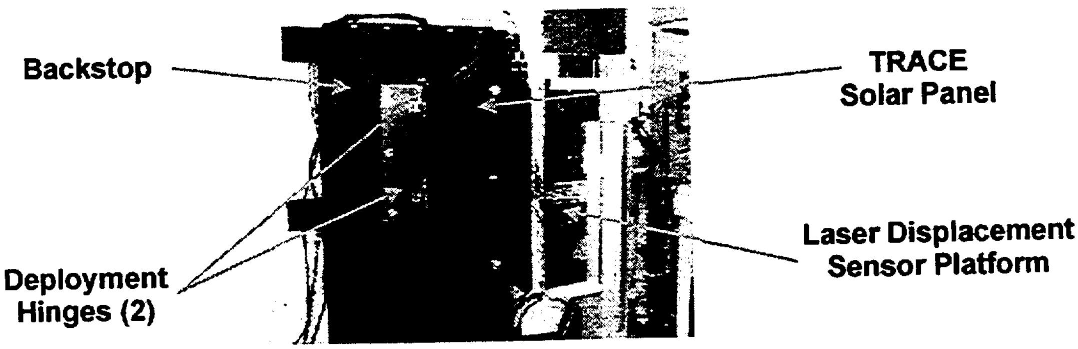


Figure 7: Photograph of test setup for TRACE solar panel thermal-structural experiments.

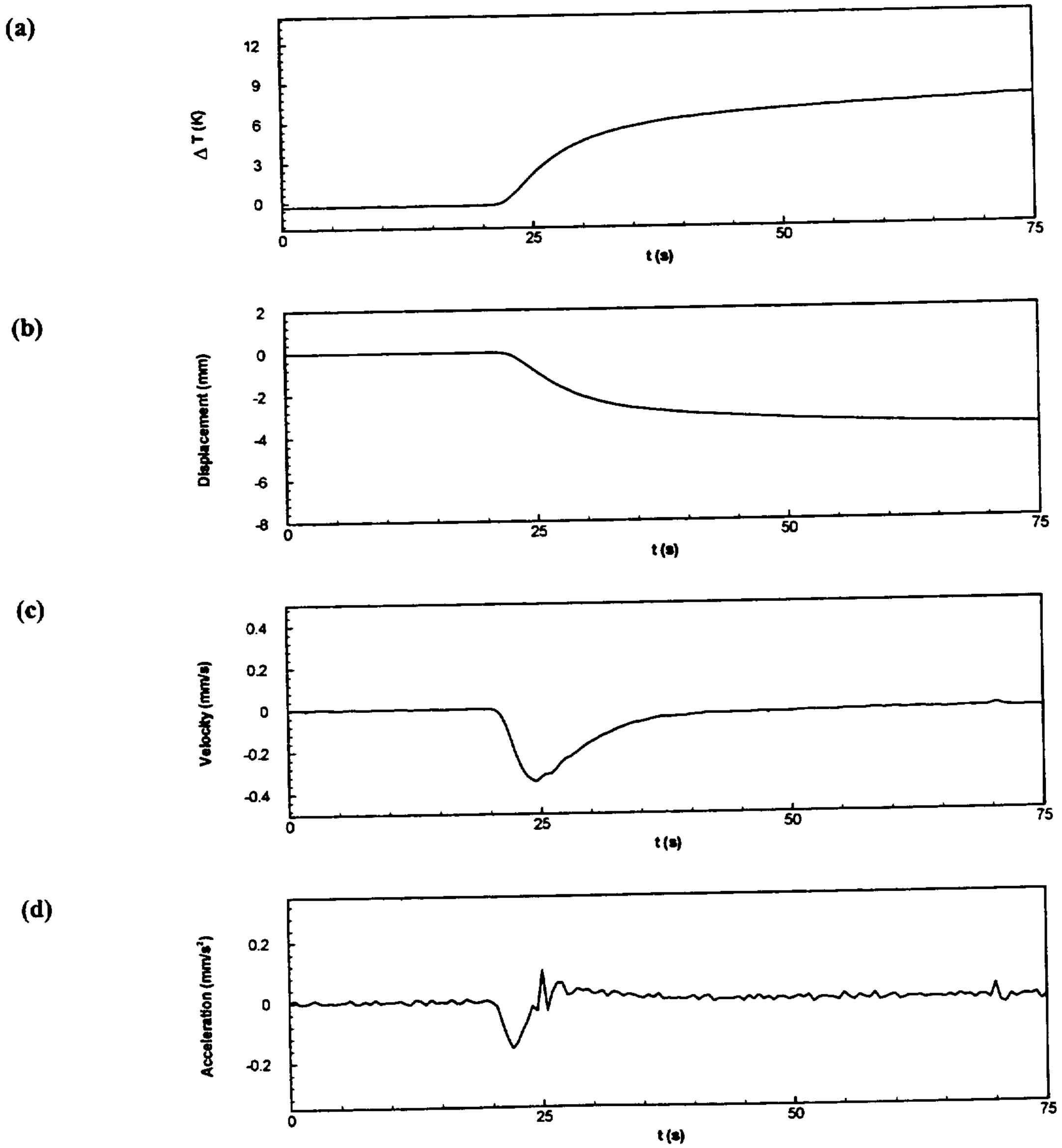


Figure 8: Representative laboratory test results for TRACE solar panel: (a) through-the-thickness temperature difference, (b) displacement at free end, (c) calculated velocity, and (d) calculated acceleration.

MIMO Fuzzy Hybrid Attitude Control for Spacecraft

Kevin Walchko[1] walchko@ufl.edu

Paul A. C. Mason[2] mason@ufl.edu

http://ice.me.ufl.edu

Abstract

This work describes the development and implementation of a Multi Input Multi Output (MIMO) hybrid fuzzy logic controller for satellites. The hybrid structure of the controller takes advantage of stabilizing effect of the classical Proportional-Derivative (PD) controller while maintaining a significant degree of robustness, performance and portability. In addition, a method is introduced for converting single output fuzzy control systems into a multi output control system, thus reducing the level of complexity in designing and implementing a MIMO fuzzy controller. A simulation study was performed using the SAMPEX satellite to validate the merits of the presented scheme. The Wie and Barber (WB) quaternion feedback controller was used for comparison in this work. During pointing and tracking maneuvers, the fuzzy provided much better performance with less control effort. The fuzzy controller also produced better performance in the presents of noise and disturbances during the pointing.

Introduction

This work deals with the development of intelligent MIMO controllers that utilize the designer's heuristic knowledge as well as any analytical description of the system to provide a reasonable control response. The proposed controller combines the PD control law with fuzzy logic. Because of its generic nature, this controller can be integrated into a variety of existing systems with minimal effort. Also, a method for simplifying the design of fuzzy controllers which reduces the complexity due to higher dimensions present in MIMO type controllers.

Background

Wie and Barber [1] developed several computationally efficient attitude control schemes for large angle maneuvers. Many of these schemes utilize quaternion and angular velocity feedback to provide stability control. Quaternions, the primary methods used for determining attitude, allow for more realistic, large angle maneuver control schemes. The control schemes developed by Wie and Barber were based on a Liapunov analysis, which defines a stable range of gains for that control law. Thus in order to meet desired performance, an ad hoc or iterative design procedure was used to achieve the desired performance. However, even when a satisfactory response was finally obtained for the nominal case, there was no guarantees how the satellite would behave in the presence of disturbances, noise, or uncertainties.

Crassidis et al [2] developed a model based nonlinear predictive control scheme for attitude control, which allowed large-angle maneuvers. This predictive control scheme, which

[1] Research Assistant, *Dept. of Mech. Eng., University of Florida, Gainesville, Fl 32611.*

[2] Assistant Professor, *Dept. of Mech. Eng., University of Florida, Gainesville, Fl 32611.* Member AIAA.

was applied in simulation to the Microwave Anisotropy Probe (MAP), determined the torque input required to make the predicted trajectories match the desired trajectories by minimizing the norm-squared error between the two. This scheme was shown to be robust against model error. Although this control scheme out performed other traditional controllers presented in the paper, such as WB, the design method was very complicated and time consuming. Thus, the design method would require an expert with in-depth knowledge to design a control system for each new satellite.

Woodard [3] developed a fuzzy controller for the Fast Auroral Snapshot Explorer (FAST) and compared it to the traditional controller (AGSS). FAST was launched in August 1996 utilizing the Attitude Ground Support System (AGSS). AGSS is a group of algorithms developed in the 1970's for the Dynamics Explorer mission. Woodard's fuzzy controller proved to be inferior to the traditional AGSS. During pointing maneuvers the fuzzy controller took 15% longer to point in the desired direction and had a larger range of errors during pointing. However Woodard did point out that fuzzy logic controller was mathematically simpler and more flexible, but not as accurate as the traditional method. "The performance of the fuzzy logic controller was slightly less desirable than that of the AGSS. This reinforces the general notion that performance with fuzzy logic controllers is sacrificed somewhat." This is an unfair notion of fuzzy logic's performance capabilities. Fuzzy logic does have the capability to achieve or surpass the performance level of other traditional control schemes, as shown in this work.

Recently, Buijtenen et al [4] developed an adaptive fuzzy logic PD controller. Their method involves a critic which predicts the future system performance and a stochastic exploration module to explore the space of possible actions. The actual adaptation process is produced by reinforcement learning. Reinforcement Learning (RL) is actually a family of biologically inspired algorithms. RL indirectly evaluates a controllers actions and rewards desired outcomes and punishes undesirable outcomes. However, their method only looks at one attitude while the method proposed here looks at the quaternions and angular velocities. Since the system is a coupled nonlinear system, changing one element of the attitude quaternion will have a dramatic effect on the other values. Also, their method relies on the critic accurately predicting several steps into the future. This is a critical component of the RL, since the critic is used in the reinforcement process. Thus if the critic is inaccurate in predicting the future, this process has no hope of properly adapting.

Outline of Paper

First, a brief description of the platform used for this study, the SAMPEX satellite, is discussed. Next, theory of the fuzzy MIMO controller, spacecraft dynamics, and stability are discussed. Then, the results of various simulations comparing the performance of the fuzzy attitude controller to the Wie and Barber controller are given. Finally, conclusions and observations are provided.

SAMPEX Satellite

SAMPEX (Solar, Anomalous, and Magnetospheric Particle Explorer) is a product of the SMEX (Small Explorer Program). This program realizes the advantages of small, quick turnaround projects. The SAMPEX satellite is designed and equipped to study the energy, composition, and charge states of particles from the explosions of supernovas, solar flares, and from the depths of interstellar space. Closer to home, SAMPEX monitors the earth's middle atmosphere as magnetospheric particle populations occasionally plunge into it. SAMPEX is a small symmetric satellite which is slightly coupled in the inertia tensor, which is shown below:

$$I = \begin{bmatrix} 15.516 & 0 & 0 \\ 0 & 21.621 & -.194 \\ 0 & -.194 & 15.234 \end{bmatrix} \cdot \mathrm{kg} - \mathrm{m}^2$$

Theory

Fuzzy Controller

The fuzzy controller, in this study, had two inputs error (e) and error velocity ($\dot{e}$) which it used to produce a single control effort (u_{fuzzy}). A schematic of the controller is shown below in Figure 1.

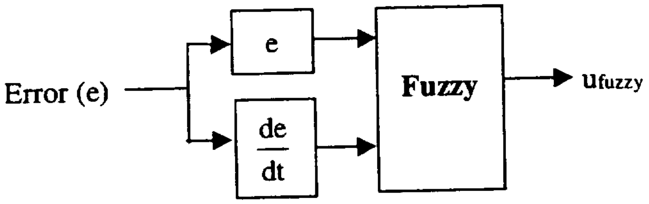


Figure 1. Fuzzy Controller

For simple systems, an engineer can easily construct a rule base for a system with two inputs based on error and one output, which defines a control effort, as shown in Figure 2. However, as the number of dimensions increase for both inputs and outputs, designing a rule base for a coupled nonlinear system grows more difficult.

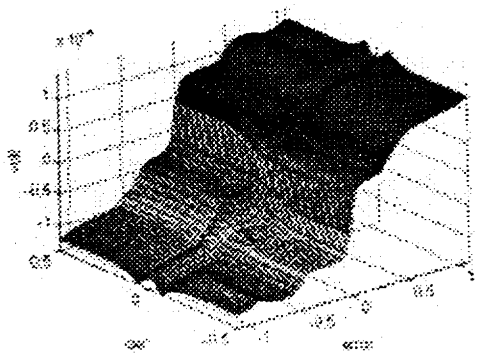


Figure 2. Control surface for the SAMPEX satellite fuzzy PD controller used in this work.

Extension of the MISO Fuzzy Hybrid Controller to MIMO Applications

In order to apply the single output fuzzy controller to a multi output problem, a technique to extend the MISO scheme into the multi output realm must be developed. This conversion (or extension) can be accomplished through the manipulation of the error vector. The error vector is used to define the direction and magnitude of the control effort to reduce the error in a minimal amount of time. The error used by the standard fuzzy PD controller is a scalar quantity. However, in the MIMO case, the error is a vector. This vector is transformed into a scalar representation by taking the norm ($|e|$) and a unit direction vector ($\bar{e}$). The scalar norm is used by the fuzzy inference system to determine control magnitude (u_{fuzzy}), which is used along with the direction to create the MIMO control vector.

$$|e| = \text{scalar representation} \Rightarrow \text{Fuzzy(scalar representation)} = u_{fuzzy}$$

$$\text{error directions} \equiv \bar{e} = \frac{e}{|e|} \quad \text{and} \quad \bar{\dot{e}} = \frac{\dot{e}}{|\dot{e}|}$$

$$\text{control vector} \equiv u_{fuzzy} \cdot (\bar{e} + \bar{\dot{e}})$$

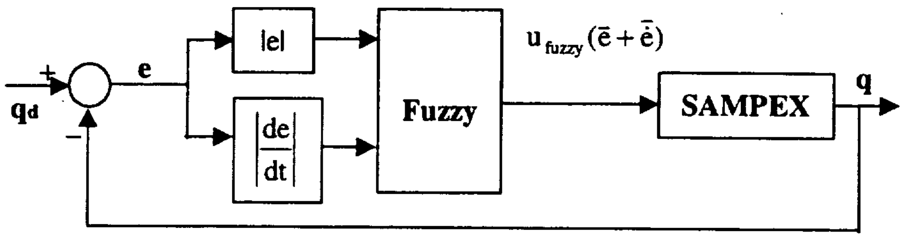

+
q_d
−
e
|e|
$\left|\frac{de}{dt}\right|$
Fuzzy
$u_{fuzzy}(\bar{e}+\bar{\dot{e}})$
SAMPEX
q

Figure 3. Fuzzy MIMO feedback system using the error vector system.

The fuzzy control structure used in this work is similar to a classical PD controller (where $T_D = 1$ and $K_P = K_D$). Thus the control effort (u_c) is:

$$u_c = u_{fuzzy}(\bar{e} + \bar{\dot{e}}) \equiv K(\bar{e} + T_D\bar{\dot{e}}) = K_P e + K_D \dot{e}$$

Spacecraft Dynamics

Next, a brief review of the kinematic and dynamic equations of motion for a three-axis stabilized spacecraft is presented. The attitude is assumed to be represented by the quaternion, defined as

$$\underline{q} \equiv \begin{bmatrix} \underline{q}_{13} \\ q_4 \end{bmatrix} \tag{1}$$

with

$$\underline{q}_{13} \equiv \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \underline{\hat{n}}\sin(\theta/2) \tag{2a}$$

$$q_4 = \cos(\theta/2) \tag{2b}$$

where $\underline{\hat{n}}$ is a unit vector corresponding to the axis of rotation and 6 is the angle of rotation. In Wertz [5] the quaternion kinematic equations of motion are,

$$\dot{\underline{q}} = \frac{1}{2}\Omega(\underline{\omega})\underline{q} = \frac{1}{2}\Xi(\underline{q})\underline{\omega} \tag{3}$$

where ω, $\Omega(\underline{\omega})$ and $\Xi(\underline{q})$ are defined as

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}=I^{-1}\begin{bmatrix}L_1\\ L_2-I_{rw}\times\omega_{rw}\\ L_3\end{bmatrix}$$

$$\Omega(\underline{\omega})\equiv\begin{bmatrix}-[\underline{\omega}\times] & \vdots & \underline{\omega}\\ \cdots\cdots & \vdots & \cdots\cdots\\ -\underline{\omega}^T & \vdots & 0\end{bmatrix} \tag{4a}$$

$$\Xi(\underline{q})\equiv\begin{bmatrix}q_4 I_{3\times3}+[\underline{q}_{13}\times]\\ \cdots\cdots\cdots\cdots\cdots\cdots\\ -\underline{q}_{13}^T\end{bmatrix} \tag{4b}$$

where $I_{n\times n}$ represent an $n\times n$ identity matrix (also, $0_{n\times m}$ will represent a $n\times m$ zero matrix), L is the angular momentum, and ω_{rw} is the angular velocity of the reaction wheels. The 3 x 3 dimensional matrices $[\underline{\omega}\times]$ and $[\underline{q}_{13}\times]$ are referred to as cross product matrices since $\underline{a}\times\underline{b}=[\underline{a}\times]\underline{b}$, with

$$[\underline{a}\times]\equiv\begin{bmatrix}0 & -a_3 & a_2\\ a_3 & 0 & -a_1\\ -a_2 & a_1 & 0\end{bmatrix} \tag{5}$$

The rotational dynamics, also known as Euler equations, are given by

$$J\dot{\underline{\omega}}=-\underline{\omega}\times(J\underline{\omega})+\underline{u}_{ext} \tag{6}$$

where J is the inertia matrix of the spacecraft, and $\underline{u}_{ext}$ is the total external torque (which includes, control torques, aerodynamic drag torques, solar pressure torques, etc.).

Stability

This section addresses the issue of stability of the controller. The stability will be examined via the Liapunov stability criteria. First, the control effort is defined as

$$u_c=-K_{P_{fuzzy}}\left(\hat{e}+\dot{\hat{e}}\right)=-K_P\cdot\Xi^T(q_{ref})q-K_D(\omega)$$

where $K_{P\_f}=K_{D\_f}$ and are constant. The variation of the fuzzy variable is lumped into the magnitude component of the error vector. The following Liapunov function is chosen.

$$V=\frac{1}{2}\omega^T\cdot J\cdot\omega+\frac{1}{2}\left(K_P(q-q_{ref})^T(q-q_{ref})\right)$$

Assuming a pointing maneuver ($\dot{q}_{ref}=0$) and taking the derivative of V yields:

$$\begin{aligned}\dot{V}&=\omega^T J\dot{\omega}+\frac{2}{2}\left(K_P(q-q_{ref})^T(\dot{q})\right)\\ &=\omega^T J\dot{\omega}+K_P(q-q_{ref})^T\dot{q}\end{aligned}$$

Now substituting in for $\dot{\omega}$ and simplifying, $\dot{V}$ reduces to:

$$\dot{V}=\omega^T(\omega\cdot x)J\omega-K_P\cdot\omega^T\Xi^T(q_{ref})q-K_D\omega^T\omega+K_Pq^T\Xi(q)-K_Pq_{ref}^T\Xi(q)\omega$$

where: $\omega^T(\omega\cdot x)=0$ $\qquad q^T\Xi(q)=0$ $\qquad \Xi^T(q)\cdot q_{ref}=-\Xi^T(q_{ref})\cdot q$

therefore

$$\dot{V}=-K_P\cdot\omega^T\Xi^T(q_{ref})q-K_D\omega^T\omega+K_Pq^T\Xi(q)\omega-K_P(\Xi^T(q)q_{ref})\omega$$

Now the last term can be transformed into:

$$+K_P(\omega^T(\Xi^T(q)q_{ref}))^T$$

Therefore we now have: $\dot{V} = -K_D\omega^T\omega$

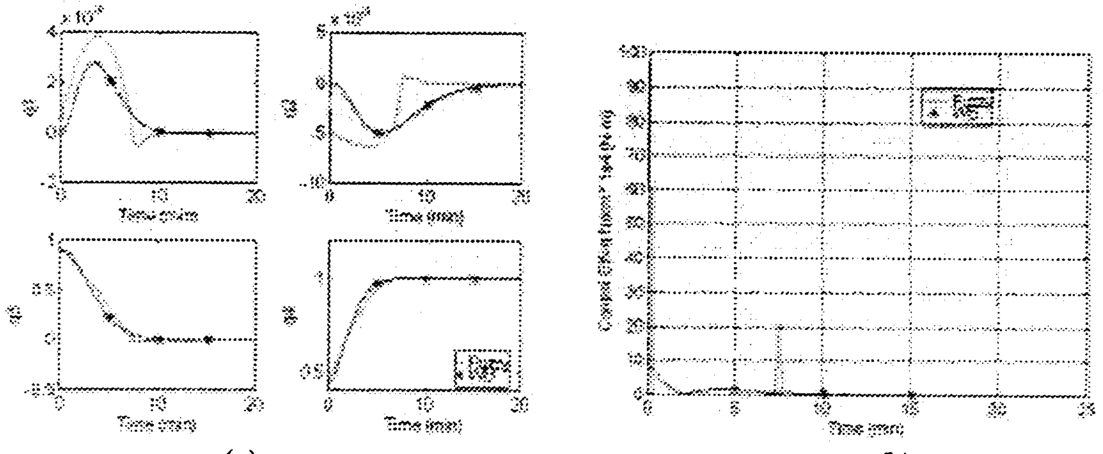
Thus, the $\dot{V}$ term will always be negative (meaning stable), if K_D is always positive. Therefore, the system is stable if K_D ($\dot{V}$ = –) is positive and K_P is positive (V = +). In the fuzzy inference system, the nonlinear mapping produces only positive K_P and K_D values, therefor, the system is stable.

Results

In order to illustrate the performance of the fuzzy MIMO hybrid controller, the SAMPEX satellite was simulated performing both a pointing and tracking maneuvers. The pointing maneuver simulated the satellite under three conditions: ideal, an impulsive disturbance, and the presence of noise. For the tracking simulation, the satellite was commanded to follow a complex maneuver. In each simulation, the satellite was initially oriented in the direction [0 0 2 1] and commanded to go to an arbitrary attitude ([0 0 0 1]) for each pointing maneuver. For each of the pointing simulations, the quaternion states, control efforts, and 3D error phase portraits are given.

Ideal Results

The satellite was commanded to reorient itself under ideal conditions (no noise or disturbances). The fuzzy controller and the WB controller produce approximately the same performance under ideal conditions as can be seen below in Figure 4. The fuzzy PD controller was able to perform the maneuver with less energy consumption than the WB controller, however, there are control effort spikes in the fuzzy PD controller that are several times larger than the WB controller. In order to quantify this, the integral of the norm of the control efforts was calculated for both controllers. The WB controller produced 275 N-m-s while the fuzzy PD controller only produced 226 N-m-s.

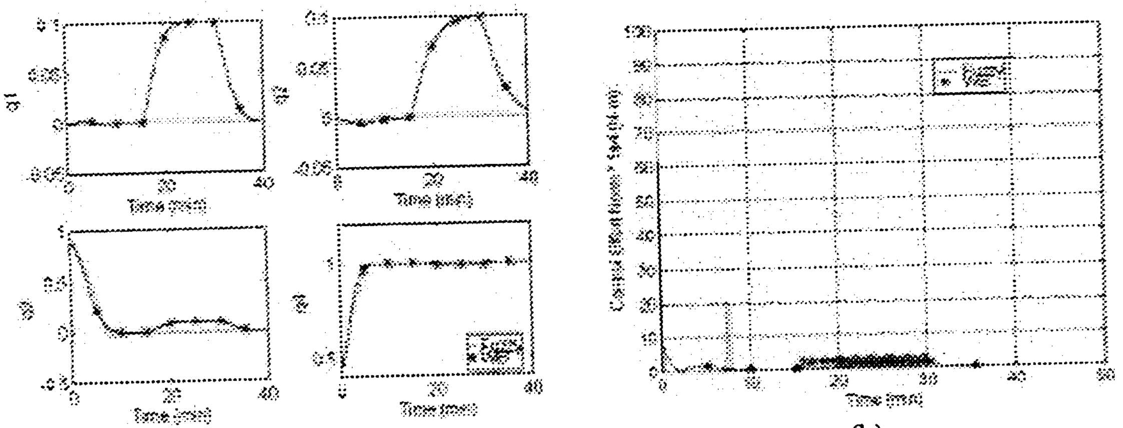


(a) (b)

Figure 4. Satellite pointing maneuver under ideal conditions.
(a) Quaternion states (b) Control Effort

Deterministic Disturbance Results

When a deterministic disturbance (.0001) was injected in to the system, it had a much greater effect on the WB controlled satellite than the fuzzy controlled satellite. Results of the simulations are shown below in Figure 5. The fuzzy PD provided better disturbance rejection and produced a lower control effort than the WB controller. The integral of the control effort for the WB controller was 847 N-m-s while the fuzzy PD was 810 N-m-s.

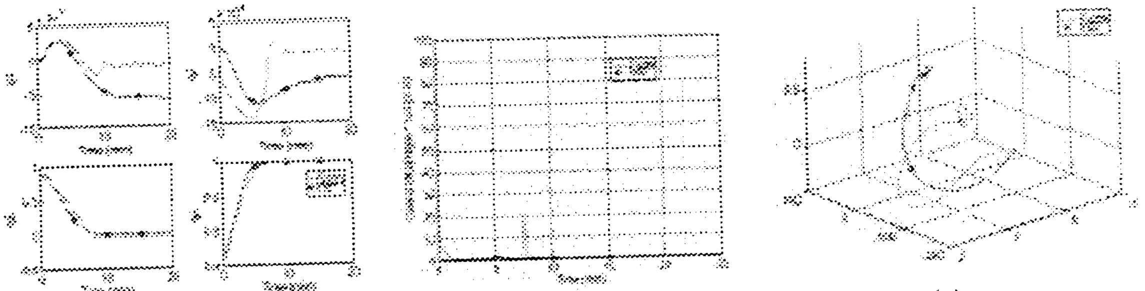


(a) (b)

Figure 5. Satellite pointing maneuver in the presents of a disturbance.
(a) Quaternion states (b) Control Effort

Noise Results

When white noise (N[.003,1])was introduced into the system, it effected the WB controller much more than the fuzzy PD. As can be seen from Figure 6, the fuzzy PD was capableof maintaining better performance, however at a higher cost. The integral of control effort for the fuzzy PD was 411 N-m-s and for WB was 290 N-m-s. This makes sense, since more energy would be needed to dampen out the effects of noise. Also, from the phase portrait, one can see how the WB controller does not converge to zero error.

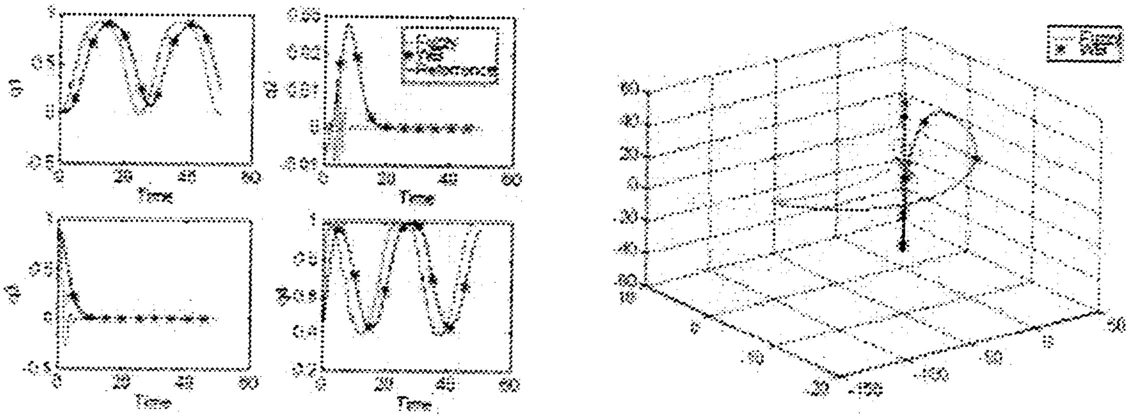


(a) (b) (c)

Figure 6. Satellite pointing maneuver in the presents of noise.
(a) Quaternion states (b) Control Effort (c) 3D error phase portrait

Tracking Maneuver

The final test of the fuzzy PD controller, was its ability to track an arbitrary path. Here the satellite was commanded to track $[.01 \cdot \sin\left(\frac{\pi}{500} t\right) 0\ 0\ 1]$[3]. The results for this simulation are shown below in Figure 7. The fuzzy PD was capable of producing much more reasonable results than did the WB controller, which was unable to perform acceptably.



(a) (b)

Figure 7. Tracking Maneuver (a) Quaternion states (b) 3D error phase portrait

Conclusions

The error vector formulations described in this study enabled a simple fuzzy controller to successfully control a complex, multi input, nonlinear system. The SAMPEX test bed produced better performance not only under ideal circumstances, but also in the presents of a disturbance and noise. The fuzzy PD was capable of producing this better performance in the presents of a disturbance while producing less control effort. The fuzzy controller also was able to produce better performance in the presents of noise, but at the cost of a much higher control effort. Finally, the fuzzy MIMO PD controller was capable of tracking a complex path which the original Wie and Barber controller was unable to do.

Acknowledgments

The authors would like to thank the personnel at NASA-Goddard Space Flight Center for their technical support.

References

[1] Wie, B., and Barber, P.M., "Quaternion Feedback for Spacecraft Large Angle Maneuvers," *Journal of Guidance*, Vol. 8, No. 3, May-June 1985, pp. 360-365.

[2] Crassidis, J.L., and Markley, F.L., "Predictive Filtering for Attitude Estimation Without Rate Sensors," *Journal of Guidance, Control and Dynamics*, Vol. 20, No. 3, May-June 1997, pp. 522-527.

[3] Since the sine term changed with time, the desired quaternion was normalized each time step to ensure that it remained at unity

[3] Woodard, Mark, "Fuzzy Open-Loop Attitude Control for the FAST Spacecraft," *AIAA*

[4] Buijtenen, W. M., Schram, Gerard, Babuska, Robert, and Verbruggen, Henk B. "Adaptive Fuzzy Control of Satellite Attitude by Reinforcement Learning," *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 2, May 1998, pp. 185-194.

[5] Wertz, James R., Spacecraft Attitude Determination and Control, Kluwer Academic Publishers, 1995.

Nonlinear Fuzzy Hybrid Control of Spacecraft

Paul A. C. Mason[1]

John L. Crassidis[2]

F. Landis Markley[3]

Abstract

This paper describes a new approach for intelligent control of a spacecraft with large angle maneuvers. This new approach, based on fuzzy logic, determines the required torque to achieve a robust, high performance attitude response. This scheme extends the robustness, performance and portability of the existing linear or nonlinear attitude controllers. Formulations are presented for attitude-control but can be extended to other applications. A simulation study, which uses the new control strategy to stabilize the motion of the Microwave Anisotropy Probe spacecraft in the presence of disturbances and saturations, demonstrates the merits of the proposed scheme.

Introduction

Due to the highly nonlinear characteristics of the governing equations and parameter uncertainties, attitude control for large angle slewing maneuvers poses a difficult problem. Because of the importance of attitude control, much effort has been devoted to the development of robust attitude control schemes for spacecraft with large angle slews. Due to the computational restrictions and the accuracy requirements these closed-loop controllers must be computational inexpensive and robust. In 1984 Wie and Barba[1] derived a number of simple control schemes using quaternion and angular velocity feedback. In addition, a Lyapunov function analysis was used to prove asymptotic stability for all cases. Other full state feedback techniques have been developed based on sliding mode (variable structure) control to deal with model uncertainty.[2] This type of control has been successfully applied for large angle maneuvers using a Gibbs vector parameterization,[3] a quaternion parameterization,[4] and a modified Rodrigues parameterization.[5]

As accuracy requirements become more stringent, researchers have turned to adaptive[6] and robust[7] techniques for better performance. For example, Schaub et. al.[6] developed an adaptive scheme that estimates external torques by tracking a Lyapunov function. This method has been shown to be very robust in the presence of spacecraft modeling errors and disturbances. Adaptive and robust schemes provide a unique method of dealing with system perturbations due to external disturbances. However, these schemes usually require a significant amount of effort

[1] Assistant Professor, *Dept. of Mech. Eng., University of Florida, Gainesville, Fl 32611.* Senior Member AIAA.

[2] Assistant Professor, *Dept. of Aero. Eng., Texas A&M University College Station, TX 77843,* Senior Member AIAA.

[3] Staff Engineer, *Goddard Space Flight Center, Code 571, Greenbelt, MD 20771.* AIAA Fellow.

for the design process and are computationally expensive. To meet the need of robustness and accuracy, researchers are now examining the use of intelligent controllers such as fuzzy logic and neural networks.

One of the most promising control methodology, which can provide robustness and design simplicity, is fuzzy logic. Fuzzy logic controllers have been successfully applied in numerous applications in the military,[8] industry[9] and research[10]. Fuzzy logic describes system dynamics through membership functions and rules rather than analytical expressions. Furthermore, it allows constraints and operator experience to be encoded into the controller without resorting to analytical descriptions. For this reason, it lends itself to the control of complex time-varying systems in the presence of disturbances and uncertainty.

In applications where model based nonlinear controllers are not sufficient, a hybrid fuzzy scheme can provide structure and performance. A hybrid controller is a combination of various types of an intelligent controller and a standard linear or nonlinear control scheme. The motivation behind developing a hybrid is to take advantage of the attributes of both controllers while accounting for their weaknesses. One of the most used hybrids is the fuzzy PID[11]. Due to their structure, hybrid schemes provide a more defined control structure while increasing the accuracy, portability, robustness, and design simplicity of the controller. Furthermore, recent research in intelligent controls has provided stability analysis[12,13] and controller design methodology.

In this paper, a new approach for the control of large angle spacecraft maneuvers is presented. This nonlinear hybrid fuzzy controller is based on the Wie and Barba[1] quaternion feedback controller. Some of the advantages of the new control scheme include: (i) design simplicity (ii) robustness with respect to model uncertainties and disturbances, and (iii) the control scheme produces unbiased control errors.

The organization of this paper proceeds as follows. First, a brief summary of the kinematics and dynamics of a spacecraft is presented. Next, a brief overview of fuzzy logic theory is presented. After which, a stability and robustness analysis, based on Wie and Barber's work, is used to validate the convergence characteristic of the hybrid fuzzy control scheme. Finally, simulation results are shown for the Microwave Anisotropy Probe (MAP) spacecraft in the presence of disturbance and saturation.

Spacecraft Dynamics

In this section, a brief review of the kinematic and dynamic equations of motion for a three-axis stabilized spacecraft is shown. The attitude is represented by the quaternion, defined as

$$\underline{q} \equiv \begin{bmatrix} \underline{q}_{13} \\ q_4 \end{bmatrix} \tag{1}$$

with

$$\underline{q}_{13} \equiv \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \underline{\hat{n}}\sin(\theta/2) \qquad\qquad q_4 = \cos(\theta/2) \tag{2}$$

where $\underline{\hat{n}}$ is a unit vector corresponding to the axis of rotation and θ is the angle of rotation. The quaternion kinematic equations of motion are derived by using the spacecraft's angular velocity ($\underline{\omega}$), given by

$$\underline{\dot{q}} = \frac{1}{2}\Omega(\underline{\omega})\underline{q} = \frac{1}{2}\Xi(\underline{q})\underline{\omega} \tag{3}$$

where $\Omega(\underline{\omega})$ and $\Xi(\underline{q})$ are defined as

$$\Omega(\underline{\omega}) \equiv \begin{bmatrix} -[\underline{\omega}\times] & \vdots & \underline{\omega} \\ \cdots\cdots & \vdots & \cdots\cdots \\ -\underline{\omega}^T & \vdots & 0 \end{bmatrix} \qquad \Xi(\underline{q}) \equiv \begin{bmatrix} q_4 I_{3\times 3} + [\underline{q}_{13}\times] \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ -\underline{q}_{13}^T \end{bmatrix} \tag{4}$$

where $I_{n\times n}$ represents an $n\times n$ identity matrix (also, $0_{n\times m}$ will represent a $n\times m$ zero matrix). The 3×3 dimensional matrices $[\underline{\omega}\times]$ and $[\underline{q}_{13}\times]$ are referred to as cross product matrices $\underline{a}\times\underline{b} = [\underline{a}\times]\underline{b}$, with

$$[\underline{a}\times] \equiv \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{5}$$

Since a three degree-of-freedom attitude system is represented by a four-dimensional vector, the quaternion components cannot be independent. This condition leads to the following normalization constraint

$$\underline{q}^T\underline{q} = \underline{q}_{13}^T\underline{q}_{13} + q_4^2 = 1 \tag{6}$$

The error quaternion of two quaternions, $\underline{q}$ and $\underline{\tilde{q}}$, is defined by

$$\delta\underline{q} \equiv \begin{bmatrix} \delta\underline{q}_{13} \\ \delta q_4 \end{bmatrix} = \begin{bmatrix} \Xi^T(\tilde{q})q \\ q^T\tilde{q} \end{bmatrix} = \underline{q}\otimes\underline{\tilde{q}}^{-1} \tag{7}$$

where the operator ⊗ denotes quaternion multiplication (see Ref. [3] for details), and the inverse quaternion is defined by

$$\underline{\tilde{q}}^{-1} = \begin{bmatrix} -\tilde{q}_1 & -\tilde{q}_2 & -\tilde{q}_3 & \tilde{q}_4 \end{bmatrix}^T \tag{8}$$

Also, if Equation (8) represents a small rotation then $\delta q_4 \approx 1$, and $\delta\underline{q}_{13}$ corresponds to half-angles of rotation. The dynamic equations of motion, also known as Euler's equations, for a rotating spacecraft are given by[14]

$$\underline{\dot{H}} = -\underline{\omega}\times\underline{H} + \underline{u}_{\text{ext}} \tag{9}$$

where $\underline{H}$ is the total system angular momentum, $\underline{u}_{\text{ext}}$ is the total external torque (which includes, control torques, aerodynamic drag torques, solar pressure torques, etc.). Also, the angular velocity form of Euler's equation can be used

$$J\underline{\dot{\omega}} = -\underline{\omega}\times(J\underline{\omega}) + \underline{u} \tag{10}$$

where J is the inertia matrix of the spacecraft, and $\underline{u}$ is the total torque.

Fuzzy Logic Control

The power of fuzzy logic is its tolerance for uncertainty and its ability to interpret information in a linguistic manner via the membership functions. In the linguistic domain the input/output relationship is defined by a set of linguistic rules and membership functions, which are used to produce the appropriate control effect. Membership functions convert crisp input into linguistic variables or visa versa. Next, if-then rules are evaluated in parallel for the fuzzified inputs. Each rule evaluation forms a fuzzy output set. The individual output sets are aggregated to form a final fuzzy output set (defuzzification). This set is defuzzified through one of several methods such as center-of-area or gravity (centroid), bisector, mean-of-maximum (mom), largest-of-maximum (lom), and smallest-of-maximum (som) [15]. The Center-of-gravity (COG) is the most common defuzzification method[16]. The entire process is depicted graphically in Figure 1.

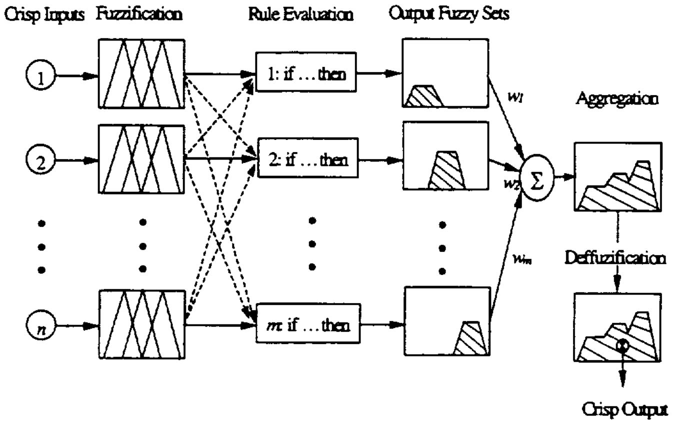


Figure 1. The Fuzzy Inference Process.

Hybrid Schemes

The most used hybrids are the fuzzy PD. In these schemes the standard feedback controller gains are replaced by a fuzzy inference system. Since the gains are a function of the error, the hybrid PD controller can be applied to any nonlinear system. The structure of the hybrid PD controller is shown in Figure 2.

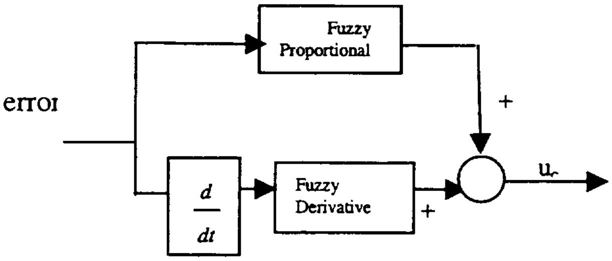


Figure 2 Hybrid Fuzzy PD Controller

Spacecraft Hybrid Fuzzy Controller

This section provides the formulation of the nonlinear fuzzy feedback controller. First, it is assumed that all of the states are available or measured.

$$\underline{y} = \underline{x} = \begin{bmatrix} \underline{q} \\ \cdots \\ \underline{\omega} \end{bmatrix} \tag{11}$$

The fuzzy hybrid structure is defined based on Wie and Barber's nonlinear quaternion feedback controller. In this fuzzy scheme the Quaternion feedback controller gains are replaced by fuzzy inference systems.

$$u = -K_p(fuzzy)\Xi^T(q_{ref})q - K_d(fuzzy)\omega \tag{12}$$

These inference systems are a nonlinear mapping that defines the controller gains based on the current states. The fuzzy inference system will improve the stability, performance and robustness by altering the gains to achieve the desired performance in the presence of saturation, disturbances, or uncertanties. The nonlinear fuzzy mapping is developed based on a priori lingusitic and analytical knowledge of the system. This information is encoded into the controller in the form of fuzzy rules. For simplicity, this fuzzy controller utilizes the norm of the quaternion error and the norm angular velocity error as inputs to the fuzzy inference system. Figures 3 and 4 provide a graphical description of the nonlinear mapping of the fuzzy inference system.

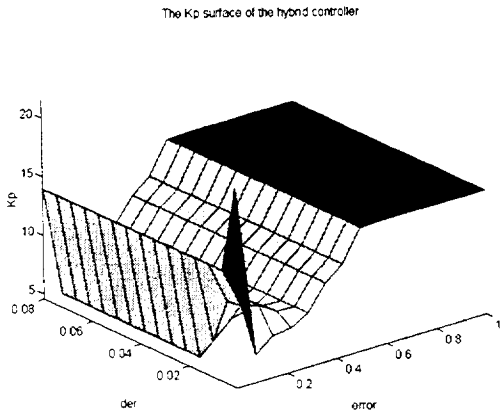


Figure 1 The nonlinear fuzzy mapping for Kp

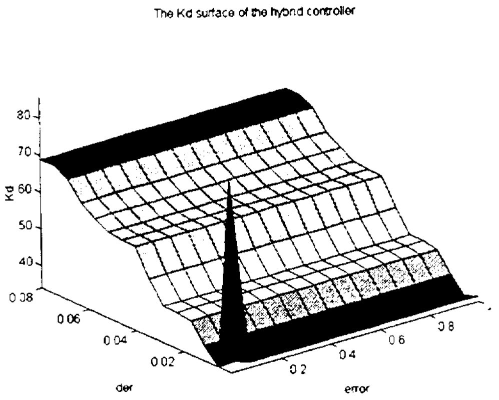


Figure 2 The nonlinear fuzzy mapping for Kd

Attitude Tracking Control for MAP

In this section, the fuzzy hybrid controller is used to control the attitude of the Microwave Anisotropy Probe (MAP) spacecraft from quaternion observations and gyro measurements. The spacecraft is due to be launched around the year 2000.

The spacecraft orbit and attitude specifications are shown in Figure 5. To provide the scan pattern, the spacecraft spins about the z-axis at 0.464 rpm, and the z-axis cones about the Sun-line at 1 rev/hour. A 22.5°±0.25° angle between the z-axis and the Sun direction must be maintained to provide a constant power input, and to provide constant temperatures for alignment stability and science quality. The instrument pointing knowledge is 1.8 arcmin (1σ), which is not required for onboard or real-time implementation.

The attitude determination hardware consists of a Digital Sun Sensor (DSS), Coarse Sun Sensors (CSS's), a star tracker, and gyroscopic rate sensors. The DSS is facing in the plus z (nominal Sun) direction. The star tracker boresight is to be pointed perpendicular to the spin axis, and 22.5° and 157.5° from the instrument apertures. The attitude control hardware includes a Reaction Wheel Assembly (RWA), which consists of three wheels oriented at a common angle to the spin axis, and distributed equally in azimuth about the spin axis. Also, the wheel torques saturate at 0.2 N-m each.

The spacecraft's attitude is defined by a 3-1-3 Euler angle rotation relative to a rotating, Sun-referenced frame. The three Euler angles are $\tilde{\phi}$, $\tilde{\theta}$, and $\tilde{\psi}$, and the desired states for observing mode are

$$\dot{\tilde{\phi}} = 1\frac{\text{rev}}{\text{hr}} = 0.001745\frac{\text{rad}}{\text{sec}} \qquad \tilde{\theta} = 22.5° = 0.3927\text{rad} \qquad \dot{\tilde{\psi}} = 0.464\text{ rpm} = 0.04859\frac{\text{rad}}{\text{sec}} \tag{1}$$

The desired Euler angles and quaternion are

$$\begin{aligned} \tilde{q}_1 &= \sin\left(\frac{\tilde{\theta}}{2}\right)\cos\left(\frac{\tilde{\phi}-\tilde{\psi}}{2}\right) & \tilde{q}_2 &= \sin\left(\frac{\tilde{\theta}}{2}\right)\sin\left(\frac{\tilde{\phi}-\tilde{\psi}}{2}\right) \\ \tilde{q}_3 &= \cos\left(\frac{\tilde{\theta}}{2}\right)\sin\left(\frac{\tilde{\phi}+\tilde{\psi}}{2}\right) & \tilde{q}_4 &= \cos\left(\frac{\tilde{\theta}}{2}\right)\cos\left(\frac{\tilde{\phi}+\tilde{\psi}}{2}\right) \end{aligned} \tag{2}$$

The kinematic equation that transforms the commanded Euler rates to the commanded body rates is given by

$$\underline{\tilde{\omega}} = \begin{bmatrix} \sin\tilde{\theta}\sin\tilde{\psi} & \cos\tilde{\psi} & 0 \\ \sin\tilde{\theta}\cos\tilde{\psi} & -\sin\tilde{\psi} & 0 \\ \cos\tilde{\theta} & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\tilde{\phi}} \\ \dot{\tilde{\theta}} \\ \dot{\tilde{\psi}} \end{bmatrix} \tag{3}$$

Two simulation studies are performed comparing the quaternion feedback (QFB) scheme with the fuzzy hybrid controller in the presence of disturbance and saturation. It should be noted that both controllers are not altered after the initial design to achieve the desired response. The controllers are designed based on a nominal system without disturbance or saturation. This aspect helps to demonstrate the robustness of the controllers.

The dominant source of disturbance for MAP is solar radiation pressure torque. The instantaneous magnitude of this torque is approximately 1×10^{-5} N-m. The spacecraft symmetry and spin will decrease the long-term average. The first simulation study involves a comparison between the fuzzy scheme and the QFB scheme in the presence of disturbances. For simulation purposes a magnitude 10 times greater than the approximate value is used. The geometric figure of the spacecraft is assumed to be a plane. A plot of the quaternion tracking errors for both controllers with a solar pressure disturbance is shown in Figure 6. Compared to the QFB controller, the fuzzy controller reduces the quaternion error with a faster settling time and smaller overshoot in the second and third error components. The fuzzy scheme also provided a smaller steady state error and a better angular velocity response (Figure 7). In addition, the fuzzy scheme achieved this type of performance with less control effort than the QFB scheme (Figure 8). The integral norm of the fuzzy controller was 19.0 and 32.0 for the QFB controller. Overall the fuzzy scheme was able to achieve a better closed-loop response than the QFB scheme with less control effort. Therefore, one can save on weight (onboard fuel) while achieving the desired performance.

In any real system there are physical limitations and constraints. These constraints usually dictate the type of controller that can be used. In some cases, additional hardware or software is used to ensure that the system remains in the feasible domain. In this work, limits are placed on the actuators to test the robustness of the fuzzy scheme in the presence of saturation. Figure 9 is a plot of the quaternion errors in the presence of saturation. As seen in Figure 9, 10, the fuzzy hybrid produced a faster damped response than the QFB scheme in the position and velocity error terms. Furthermore, the fuzzy scheme required less control authority to achieve this response (Figure 11). The integral norm of the fuzzy controller is 18.8 and 19.6 for the QFB controller. Because of the chosen limits, only the third component of the control effort saturates. The saturation of the controller reduces the speed of convergence and can cause other errors. As seen in Figure 12, the error convergence in the presence of saturation is not optimal.

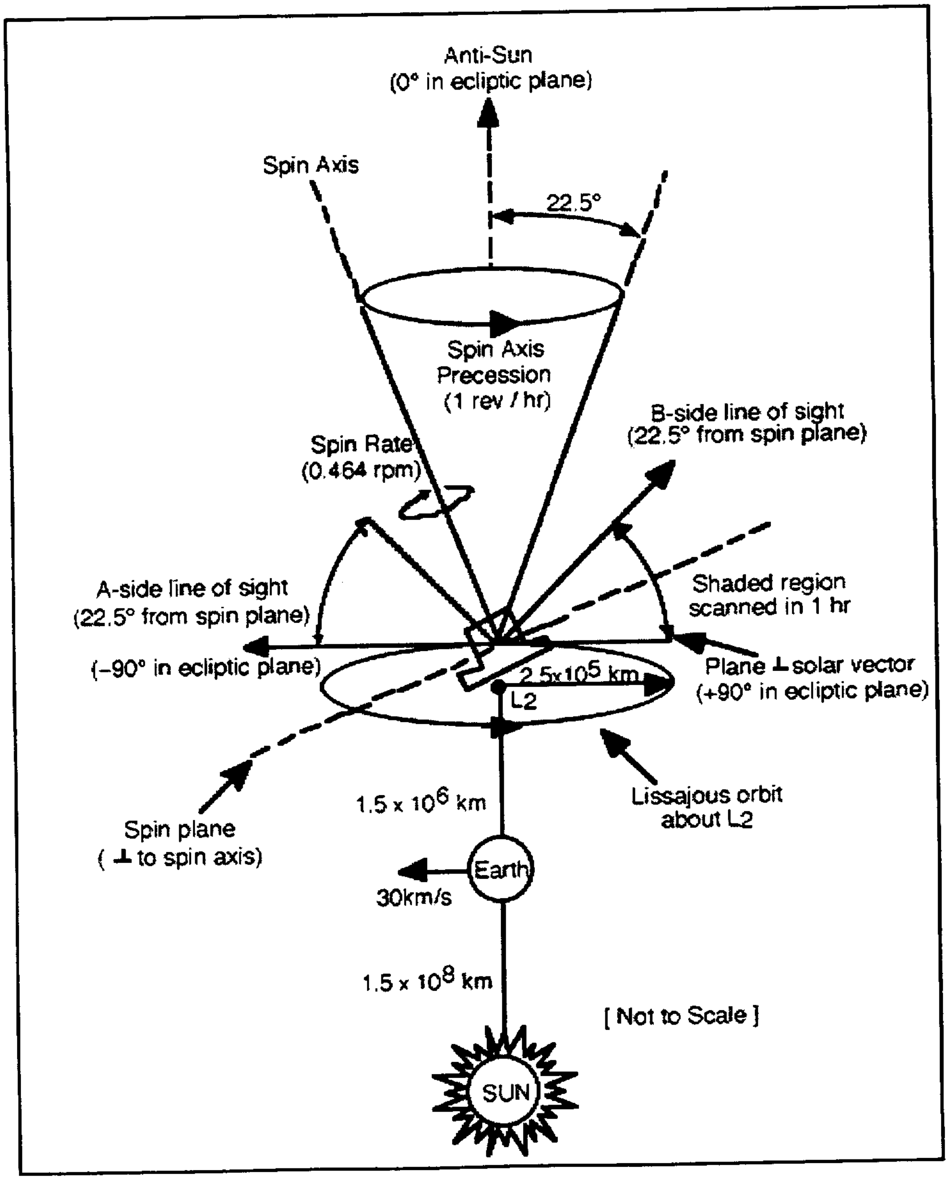


Figure 5 MAP Spacecraft Specifications

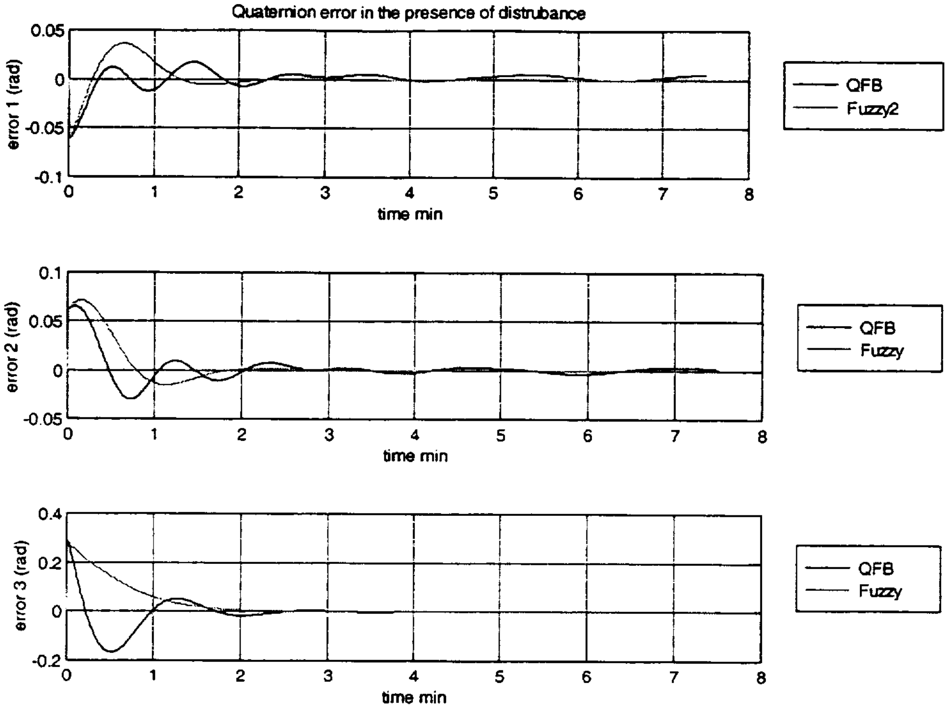


Figure. 6 Quaternion errors in the Presence of Disturbance

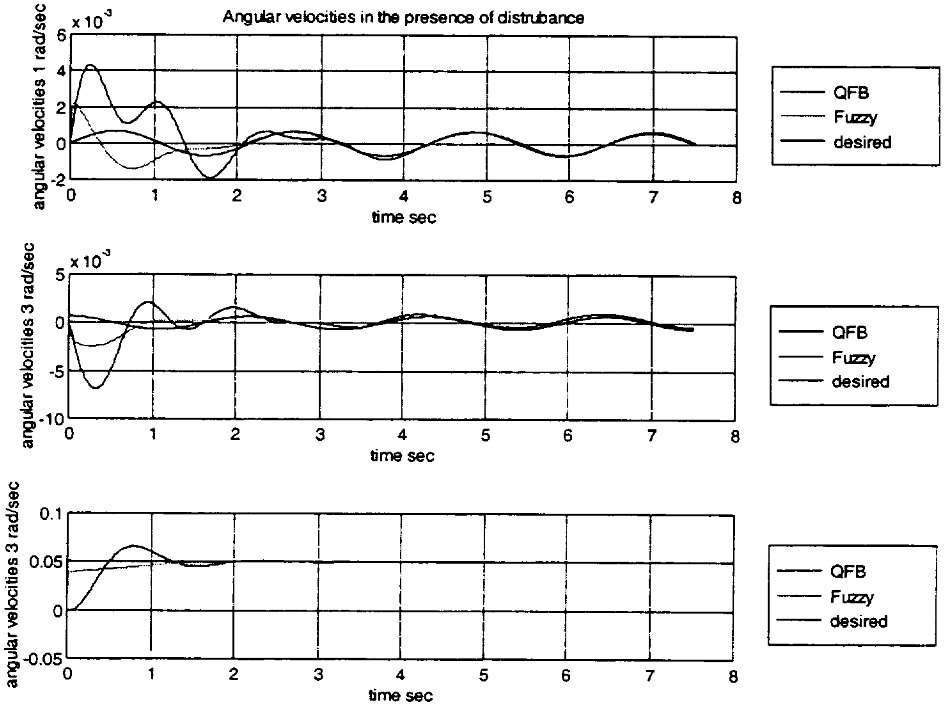


Figure. 7 Angular Velocities of in the Presence of Disturbances

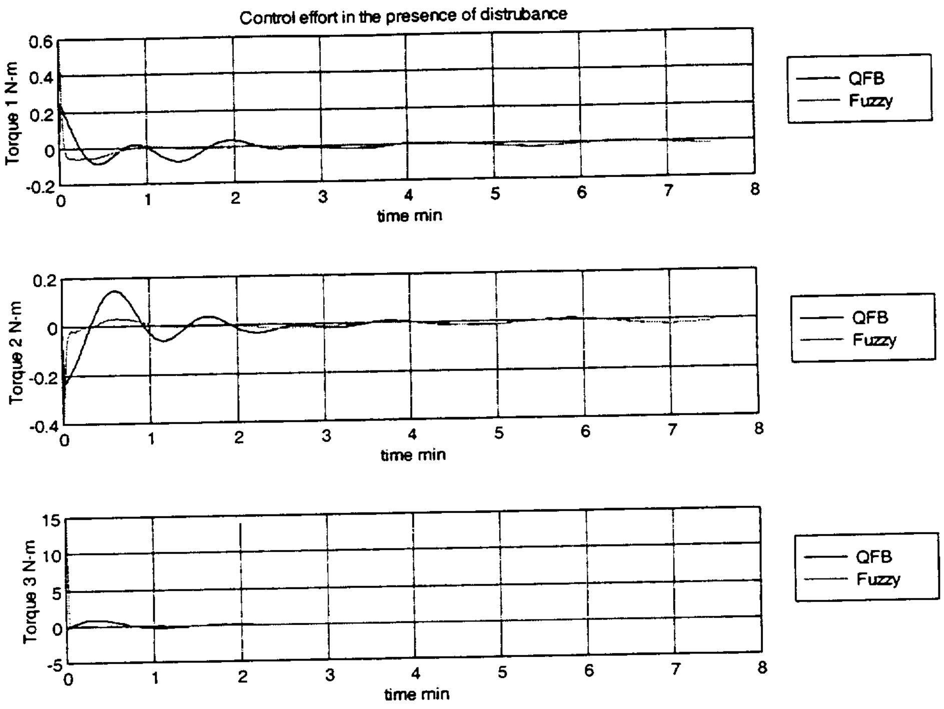


Figure. 8 Control effort in the Presence of Disturbances

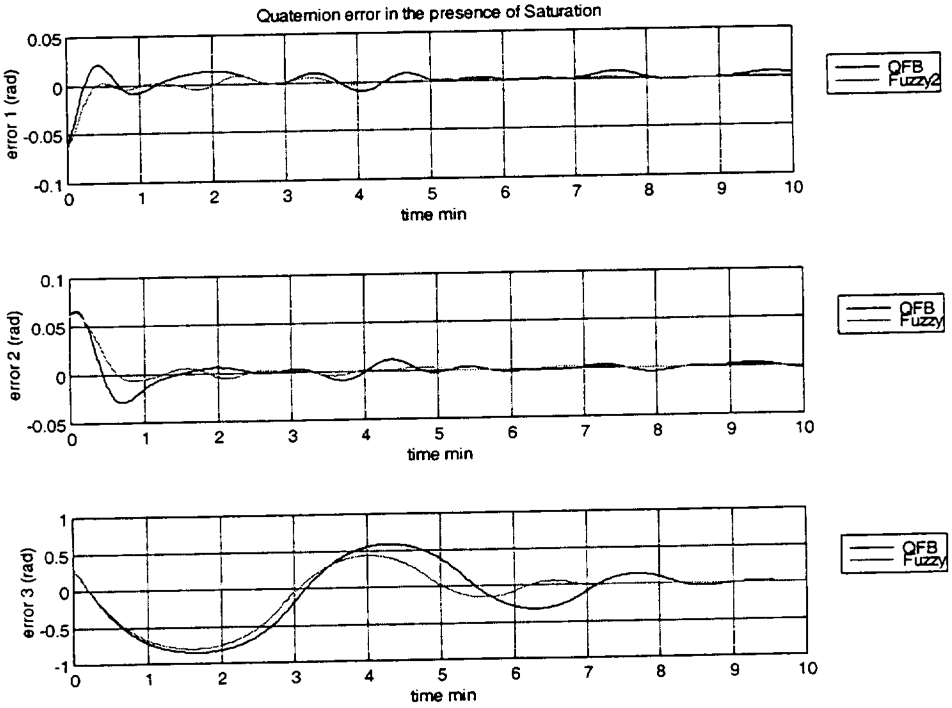


Figure. 9 Quaterion error in the presence of saturation

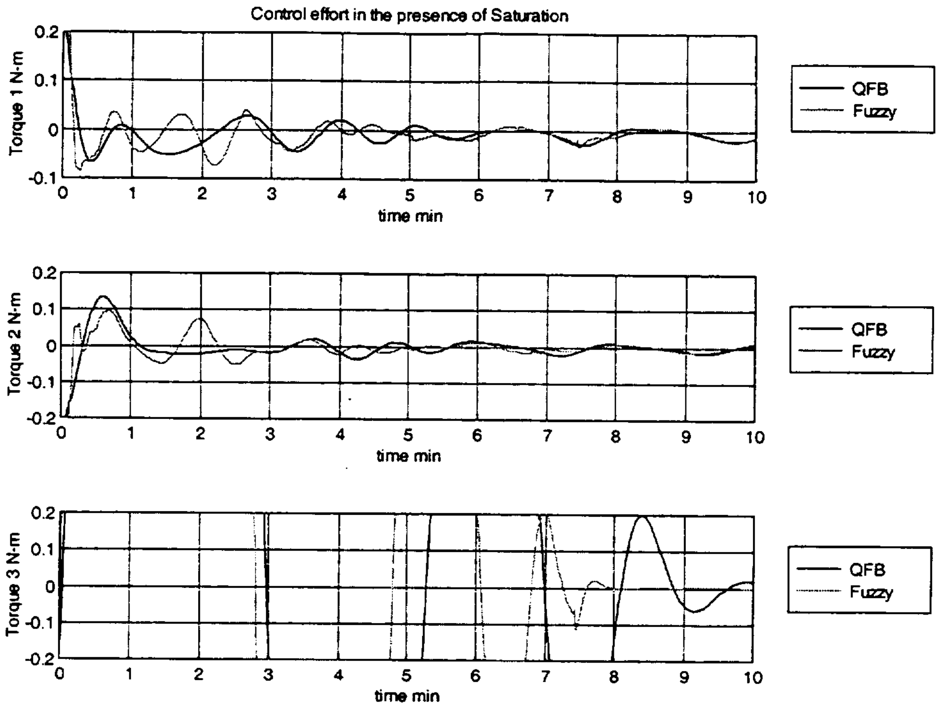


Figure. 10 Angular Velocities of in the Presence of Saturation

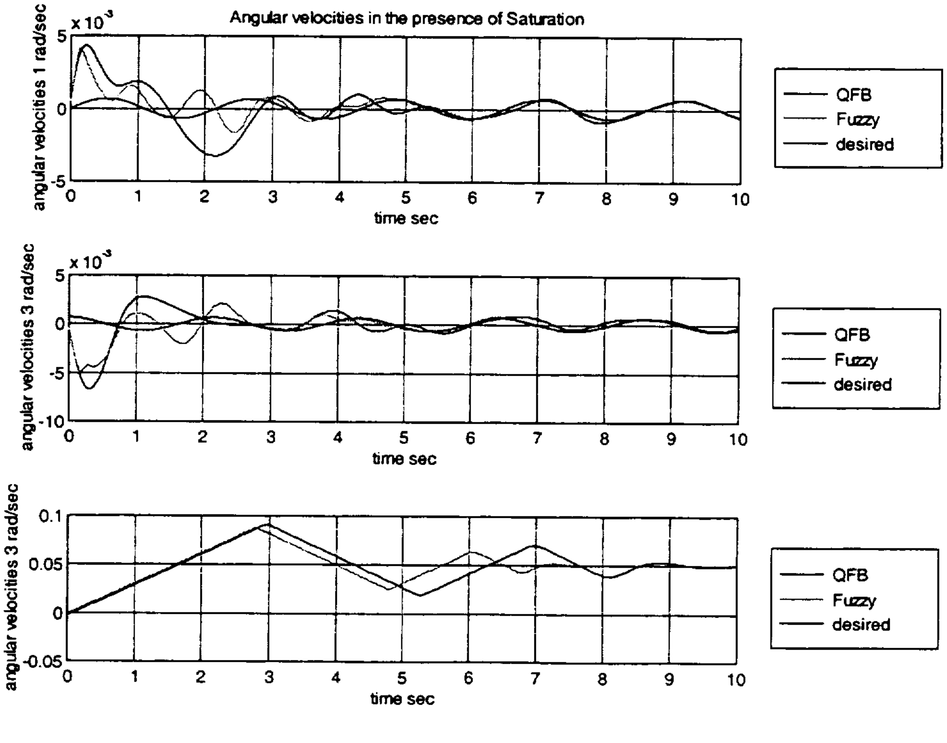


Figure. 11 Control effort in the presence of saturation

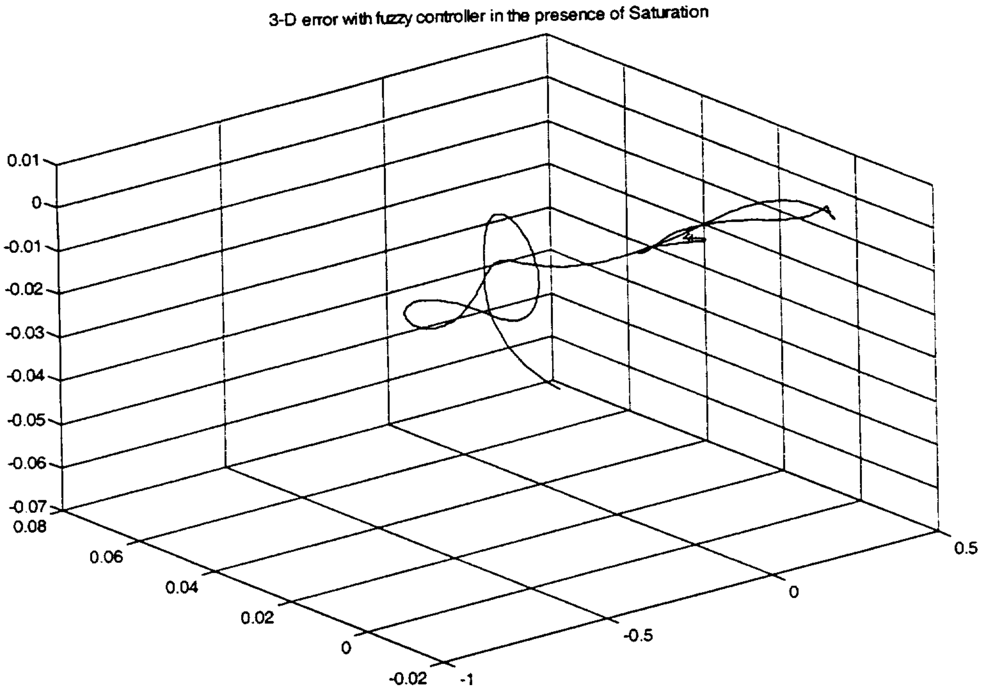


Figure 12 Error Phase Portrait in the presence of saturation

Conclusions

In this paper, a new approach for the control of a spacecraft with large angle maneuvers was presented. The presented work is a preliminary study, and illustrates how a simple hybrid fuzzy scheme can provide robustness while maintaining a high degree of simplicity. The new approach was developed using a fuzzy logic to define the control torques. A simulation study was shown comparing the new controller with a more traditional proportional-derivative type controller for the Microwave Anisotropy Probe spacecraft. Figures 6-8 illustrate the fuzzy controller ability to compensate for disturbance torques more effectively than the QFB controller. In addition, Figures 8-12 provided an indication of the schemes robustness in the presence of saturation. Results of this simulation study indicate that the fuzzy hybrid controller converges to the desired values faster than the QFB controller, and provides robustness in the presence of disturbances and saturation.

Acknowledgments

The first author's work was partially supported by a ASEE/NASA Fellowship tenured at NASA-Goddard Space Flight Center. The author greatly appreciates this support.

References

1. Wie, B., and Barba, P.M., "Quaternion Feedback for Spacecraft Large Angle Maneuvers," Journal of Guidance, Vol. 8, No. 3, May-June 1985, pp. 360-365.
2. Slotine, J.J.E., and Li, W., Applied Nonlinear Control, Prentice Hall, NJ, 1991.

3. Ramirez, H.S., and Dwyer, T.A.W., "Variable Structure Control of Spacecraft Reorientation Maneuvers," Proceedings of AIAA Guidance, Navigation, and Control Conference, Williamsburg, VA, Aug. 1986, AIAA Paper #86-1987, pp. 88-96.
4. Vadali, S.R., "Variable Structure Control of Spacecraft Large Angle Maneuvers," Journal of Guidance, Control and Dynamics, Vol. 9, No. 2, March-April 1986, pp. 235-239.
5. Crassidis, J.L., and Markley, F.L., "Sliding Mode Control Using Modified Rodrigues Parameters," Journal of Guidance, Control and Dynamics, Vol. 19, No. 6, Nov.-Dec. 1996, pp. 1381-1383.
6. Schaub, H., Junkins, J.L., and Robinett, R.D., "Adaptive External Torque Estimation By Means of Tracking a Lyapunov Function," AAS/AIAA Space Flight Mechanics Meeting, Austin, TX, Feb. 1996, AAS Paper #96-172.
7. Kang, W., "Nonlinear H∞ Control and Its Application to Rigid Spacecraft," IEEE Transactions on Automatic Control, Vol. 40, No. 7, July 1995, pp. 1281-1285.
8. Gonsalves, P., G., and Caglayan, A., K., "Fuzzy Logic PID Controller for Missile Terminal Guidance," IEEE International Symposium on Intelligent Control - Proceedings, 1995, pp. 377-382.
9. Bartolini, G., Casalino, G., Davoli, F., Mastretta, M., and Morten, E., "Development of Performance Adaptive Fuzzy Controllers with Application to Continuous Casting Plants," Industrial Applications of Fuzzy Control, Edited by Sugeno, M., Elsevier Science Publishers, North Holland, 1985.
10. Lee, J.X.; G. Vukovich; and J.Z. Sasiadek. "Fuzzy Control of A Flexible Link Manipulator", Proceedings of the American Control Conference, June 1994, 568-574.
11. Wang, H. O., Tanaka, K., Griffin, M.. "An Approach to fuzzy Control of Nonlinear Systems: Stability and Design Issues," *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 1, Feb. 1996, pp. 14-23.
12. Spooner, J., and Passino, K. M., "Stable Adaptive Control Using Fuzzy Systems and Neural Networks," *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 3, Aug. 1996, pp. 339-359.
13. Brehm, T., Kuldip, S., R., "Hybrid Fuzzy Logic PID Controller," *IEEE International Conference on Fuzzy Systems*, Vol. 3, 1994, pp. 1682-1687.
14. Wertz, J.R. (editor), *Spacecraft Attitude Determination and Control*, Wertz, , Kluwer Academic Publishers, Dordrecht, 1978.
15. Gulley, Ned, Roger Jang, J. S. Fuzzy Logic Toolbox for use with Matlab®. The MathWorks, Inc., 1995.
16. Kosko, B. Fuzzy Engineering. New Jersey: Prentice Hall, 1997.
17. Lu, P., "Constrained Tracking Control of Nonlinear Systems," *System and Control Letters*, Vol. 27, 1996, pp. 305-314.

COMPARISON OF FOUR TORQUE DISTRIBUTION METHODS FOR ATTITUDE CONTROL

Dr. Mark E. Pittelkau
Orbital Sciences Corporation
pittelkau.mark@orbital.com

ABSTRACT

Torque distribution is useful in attitude control systems where continuous momentum control is desired or where there are fewer than three reaction wheels. Torque distribution algorithms divide the total commanded control torque between the reaction wheels and the magnetic torque rods or coils. The attitude control systems in four satellites—OrbComm, COMET, SSTI/CLARK, and TSX-5—were each designed with different torque distribution algorithms. Three of these algorithms use a least-squares projection while the fourth, a new method, uses a direct projection of commanded torques onto the subspaces spanned by the actuators. The relative merits of these methods will be shown by analysis. The new method is implemented in TSX-5 and will be flown in 1999. A clear mathematical representation of the subspace projections for this method and simulation results are presented.

INTRODUCTION

Control actuators for spacecraft attitude control systems generally comprise reaction wheels and magnetic torque rods or coils. Together these are redundant torque actuators, but system designs often do not take full advantage of this redundancy. Most often the reaction wheels are made to control attitude in an inner loop and the torque rods are made to reduce wheel momentum in an outer loop. This is termed "momentum management," which is a poor name because momentum control is an intrinsic part of the closed-loop dynamics of the spacecraft.

The optimal control method presented in ref. 1 was for a momentum bias spacecraft but is readily extended to two, three, and four wheel systems. This approach provides the optimal trade of actuator control effort between reaction wheels and magnetic torque rods. The frequency-weighted LQG control in ref. 2 divides the control effort between the torque rods and the pitch reaction wheel to control pitch attitude and momentum. Optimal solutions to attitude and momentum control are derived in ref. 3 using cost functions. Performance results are presented in that paper.

The torque distribution methods of Grossman (ref. 4, 5) and Krebs (ref. 6) utilize an overdetermined set of equations that relate the actuator commands to the commanded attitude control torque and momentum control torque and includes a constraint on the direction of the magnetic dipole. This set of equations is then solved by using QR decomposition and backward substitution, which is a least-squares solution. This approach has been successfully used on OrbComm (ref. 6, 7) and was applied to the ACS design on COMET (ref. 5) and CLARK (ref. 4).

This paper presents a new torque distribution algorithm that divides the commanded control torque between the reaction wheels and the magnetic torque rods or coils so that the wheels produce only the torque that cannot be produced magnetically. One distinction between this and previous methods is that in this method the distribution is obtained by a direct projection of commanded torques onto the subspaces spanned by the actuators, rather than by solving a matrix equation. Other distinctions are that the torque is truely distributed and that the distribution can be adjusted by a scalar parameter. It is shown in this paper that this approach is the solution to the system of equations in ref. 4. Other characteristics of the previous methods are revealed as well. The Direct Subspace Projection method is illustrated with simulation results designed to demonstrate the effect of the subspace projections on actuator effort.

Nomenclature and notation are summarized in appendix A in case the reader needs a quick reminder of the definition of a symbol. These brief definitions also refer to the equation where the symbol is defined.

TORQUE DISTRIBUTION METHODS

Torque Distribution on CLARK

The torque distribution method developed for CLARK (ref. 4) was designed to address the problem of redundancy in a three-wheel system where one of the wheels may fail. The torque distribution equations on CLARK relate the actuator commands to the desired attitude control torque and wheel momentum control torque. These equations in matrix form are

$$\begin{bmatrix} \tau_{att} \\ \tau_h \\ 0 \end{bmatrix} = \begin{bmatrix} I & -[B\times] \\ 0 & -[B\times] \\ 0 & B^T \end{bmatrix} \begin{bmatrix} \tau_w \\ D \end{bmatrix} \tag{1}$$

where B is the geomagnetic field vector, τ_{att} is the commanded attitude control torque, and where τ_h is the momentum control torque. The last equation is a constraint on the direction of the magnetic dipole D. The 7×6 matrix is full column rank and therefore can be solved by using QR decomposition and backward substitution. This is simply a least-squares solution that yields the commanded wheel torque τ_w and magnetic dipole moment D.

In the case of a failed wheel, the roll (x) wheel for example, the equations are modified to force the roll wheel torque to zero. The roll wheel momentum control torque equation is also eliminated (implicitly $\tau_{hx} = 0$). Thus

$$\begin{bmatrix} \tau_{attx} \\ \tau_{atty} \\ \tau_{attz} \\ 0 \\ \tau_{hy} \\ \tau_{hz} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & B_z & -B_y \\ 0 & 1 & 0 & -B_z & 0 & B_x \\ 0 & 0 & 1 & B_y & -B_x & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -B_z & 0 & B_x \\ 0 & 0 & 0 & B_y & -B_x & 0 \\ 0 & 0 & 0 & B_x & B_y & B_z \end{bmatrix} \begin{bmatrix} \tau_{wx} \\ \tau_{wy} \\ \tau_{wz} \\ D_x \\ D_y \\ D_z \end{bmatrix} \tag{2}$$

Torque Distribution on COMET

The torque distribution method developed for COMET (ref. 5) was designed to address the problem of three-axis torque control with only two wheels and three magnetic torque coils.

The wheels are aligned with the y and z body axes. The approach is somewhat similar to that presented in the previous section. A significant difference is the use of a "momentum management" axis defined by

$$n = \frac{x \times b}{|x \times b|} \qquad b = B/|B| \,, \tag{3}$$

which is the axis at the intersection of the plane of all possible magnetic torques and the plane of all possible wheel torques. Wheel momentum can be controlled only in this direction without disturbing the spacecraft. The attitude control torque τ_{att} is divided into two parts, the projection τ_1 onto the space spanned by nn^T and the projection τ_2 onto the orthogonal complement of the space spanned by nn^T. These are written

$$\tau_1 = nn^T\tau \tag{4}$$

$$\tau_2 = (I - nn^T)\tau \,. \tag{5}$$

The torque τ_2 comprises a magnetic torque τ_{m1} and a wheel torque τ_{w1}, which are computed as the solution to the overdetermined matrix equation

$$\begin{bmatrix} 0 \\ 0 \\ \tau_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} b^T & 0 \\ 0 & x^T \\ I - bb^T & I - xx^T \\ n^T & 0 \\ 0 & n^T \end{bmatrix} \begin{bmatrix} \tau_{m1} \\ \tau_{w1} \end{bmatrix} \tag{6}$$

and τ_1 is given by

$$\tau_1 = \begin{bmatrix} nn^T & nn^T \end{bmatrix} \begin{bmatrix} \tau_{m1} \\ \tau_{w1} \end{bmatrix} \,. \tag{7}$$

These equations say that τ_{m1} is perpendicular to B, τ_{w1} is has no component along the x-axis, that τ_2 comprises a magnetic torque and a wheel torque, and that τ_{m1} and τ_{w1} are perpendicular to the n-axis or are zero (see ref. 5). A momentum control torque is projected onto the n-axis and added to τ_{m1} and subtracted from τ_{w1} to get the total magnetic and wheel torques.

Examination of these equations shows an inconsistency in the last two constraints: τ_1 is zero if τ_{m1} and τ_{w1} are perpendicular to the n-axis. Equations (4) and (7) are corrected versions of equations (8) and (11) in ref. 5, which reads $\tau_1 = n^T\tau$ and $\tau_1 = [n^T \; n^T]\left[\begin{smallmatrix} \tau_{m1} \\ \tau_{w1} \end{smallmatrix}\right]$. Equations (6) and (7) (equations (10) and (11) in ref. 5) probably should have been written

$$\begin{bmatrix} 0 \\ 0 \\ \tau_2 \\ \tau_1 \end{bmatrix} = \begin{bmatrix} b^T & 0 \\ 0 & x^T \\ I - bb^T & I - xx^T \\ nn^T & nn^T \end{bmatrix} \begin{bmatrix} \tau_{m1} \\ \tau_{w1} \end{bmatrix} \,. \tag{8}$$

Equations (6) and (8) become ill-conditioned as the direction of the magnetic field aligns with the x-axis, resulting in large control torques. The n-axis in (3) also becomes poorly defined. This is because a large magnetic dipole is required to produce torque about the x-axis as b approaches x. This can easily saturate the magnetic torque bars and the wheels. Equations (6) and (8) are solved by QR decomposition and backward substitution.

Torque Distribution on OrbComm

The OrbComm spacecraft (ref. 6, 7) was originally designed to operate only with magnetic torque coils, but performance and stability would be poor over the poles where no magnetic yaw control is available. A very small reaction wheel was added to the control system to improve performance and stability. The torque distribution law reported in ref. 6 is similar to that used on COMET except that the n-axis is not used. This law is written

$$\begin{bmatrix} b_{1x} & b_{2x} & 0 \\ b_{1y} & b_{2y} & 0 \\ b_{1z} & b_{2z} & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_{m1} \\ \tau_{m2} \\ \tau_w \end{bmatrix} = \begin{bmatrix} \tau_{attx} \\ \tau_{atty} \\ \tau_{attz} \\ \tau_h \end{bmatrix} \tag{9}$$

where b_1 and b_2 are basis vectors for the plane perpendicular to the magnetic field vector, τ_{m1} and τ_{m2} are magnetic torques in that plane, and τ_w is the wheel torque.[1] The magnetic torque along the body axes and wheel torque are given by the least squares solution

$$\begin{bmatrix} \tau_m \\ \tau_w \end{bmatrix} = A(A^T A)^{-1} A^T \begin{bmatrix} \tau_{att} \\ \tau_h \end{bmatrix} , \tag{10}$$

where A is the coefficient matrix in (9). This equation is solved by QR decomposition.

This algorithm suffers from a singularity (ref. 6) as does the COMET algorithm, so it was suggested in ref. 6 to augment the A matrix with an cost weighting on the actuator commands τ_c and τ_h. This weighting is simply $K\left[\begin{smallmatrix}\tau_m \\ \tau_w\end{smallmatrix}\right]$ where K is a 3×3 diagonal matrix.

TORQUE DISTRIBUTION BY DIRECT SUBSPACE PROJECTION

The torque distribution methods in ref. 4, 5, 6 and summarized in the previous sections were derived from the concept of mapping the actuator torques to the body axes in which the commanded torque is represented and then solving for the actuator torques. These methods are termed *indirect*. The new concept presented in this paper is direct and works the other way around. The commanded torque is projected onto orthogonal subspaces defined by the magnetic field vector: the subspace $\mathcal{B}$ containing vectors in the direction of B and the subspace $\mathcal{B}^\perp$ containing vectors perpendicular to B. Magnetic control torque is generated for any vector in $\mathcal{B}^\perp$ and any other control torque must be in $\mathcal{B}$ and generated by reaction wheels, otherwise an error torque will be generated.

This torque distribution method is called the Direct Subspace Projection method because the available control torque space is decomposed into orthogonal complements and the desired control torque is projected directly onto these spaces to obtain the commanded wheel and magnetic torque. This method is used in the attitude controller for TSX-5, which will be launched in 1999. A few definitions are in order before proceeding with the development of the Direct Subspace Projection method.

[1] The notation used here is modified a bit from what is in the original paper to be consistent with notation previously used.

Definitions

The concepts presented in this paper are based upon the theory of vector spaces (ref. 8), in particular the Euclidean space $\boldsymbol{R}^n$, the set of all $n \times n$ matrices of real numbers. Projection matrices are a special subspace of $\boldsymbol{R}^n$. In this work we are concerned mainly with the space $\boldsymbol{R}^3$ and so the projection matrices are of dimension 3 × 3. A *projection* is a particular mapping of a vector from one Hilbert space onto another. A *projector* is a linear operator that performs this mapping. For example, if $\mathcal{S} \subset \boldsymbol{R}^3$ ($\mathcal{S}$ is a subspace of $\boldsymbol{R}^3$) then $P_{\mathcal{S}} : \boldsymbol{R}^3 \mapsto \mathcal{S}$ ($P_{\mathcal{S}}$ maps vectors in $\boldsymbol{R}^3$ onto $\mathcal{S}$). The *domain* of $\mathcal{S}$ is $\boldsymbol{R}^3$ and $\mathcal{S}$ is its *range*. Let $w \in \boldsymbol{R}^3$ (w is a vector in $\boldsymbol{R}^3$). Then a projection of w onto $\mathcal{S}$ is $u = P_{\mathcal{S}}w$. The *orthogonal complement* $\mathcal{S}^{\perp}$ of a space $\mathcal{S}$ is such that for any $u \in \mathcal{S}$ with $u \neq 0$, $u \notin \mathcal{S}^{\perp}$. It follows that $\mathcal{S} \cap \mathcal{S}^{\perp} = \{0\}$ and $\mathcal{S} \cup \mathcal{S}^{\perp} = \boldsymbol{R}^3$. Thus $P_{\mathcal{S}}u = u$ and $P_{\mathcal{S}^{\perp}}u = 0$, so $P_{\mathcal{S}}P_{\mathcal{S}^{\perp}} = 0$ and $P_{\mathcal{S}^{\perp}}P_{\mathcal{S}} = 0$. It follows that the eigenvalues of a projector are either 0 or 1. It also follows that projectors are idempotent, i.e., $P_{\mathcal{S}}P_{\mathcal{S}} = P_{\mathcal{S}}$ and $P_{\mathcal{S}^{\perp}}P_{\mathcal{S}^{\perp}} = P_{\mathcal{S}^{\perp}}$. Projectors are not commutative, so for example $P_{\mathcal{B}}P_{\mathcal{A}} \neq P_{\mathcal{A}}P_{\mathcal{B}}$, where $P_{\mathcal{A}}$ and $P_{\mathcal{B}}$ are different projectors. It can be shown that projectors are self-adjoint, i.e., $P_{\mathcal{S}}^T = P_{\mathcal{S}}$ and $P_{\mathcal{S}^{\perp}}^T = P_{\mathcal{S}^{\perp}}$. Any vector $w \in \boldsymbol{R}^3$ can be written as the sum $w = u + v$ of orthogonal vectors ($u \perp v$), where $u = P_{\mathcal{S}}w$ and $v = P_{\mathcal{S}^{\perp}}w$. It follows that $P_{\mathcal{S}} + P_{\mathcal{S}^{\perp}} = I$, the identity matrix, and $P_{\mathcal{S}^{\perp}} = I - P_{\mathcal{S}}$. The *range space* $\mathcal{R}(P)$ of a projector P is the set of all vectors that result from the mapping of the domain $\mathcal{D}(P)$ of the projector onto its range. Let $u,\ v \in \boldsymbol{R}^3$. Then using the definitons above, $u \in \mathcal{R}(P_{\mathcal{S}})$ and $w - u \in \mathcal{R}(P_{\mathcal{S}^{\perp}})$. Also $\mathcal{R}(P_{\mathcal{S}}) = \mathcal{S}$ and $\mathcal{R}(P_{\mathcal{S}^{\perp}}) = \mathcal{S}^{\perp}$ since the domain of $\mathcal{D}(P_{\mathcal{S}}) = \boldsymbol{R}^3$ and $\mathcal{D}(P_{\mathcal{S}^{\perp}}) = \boldsymbol{R}^3$.

Let $\mathcal{B} = \{B\}$ be the subspace containing the magnetic field vector B. The orthogonal projectors $P_{\mathcal{B}}$ and $P_{\mathcal{B}^{\perp}}$ are given by

$$P_{\mathcal{B}} = bb^T \qquad\qquad P_{\mathcal{B}^{\perp}} = I - bb^T \tag{11}$$

where $b = B/|B|$. These projectors are orthogonal ($P_{\mathcal{B}}P_{\mathcal{B}^{\perp}} = 0$ and $P_{\mathcal{B}^{\perp}}P_{\mathcal{B}} = 0$), their sum spans $\boldsymbol{R}^3$ ($P_{\mathcal{B}} + P_{\mathcal{B}^{\perp}} = I$), and they are idempotent ($P_{\mathcal{B}}P_{\mathcal{B}} = P_{\mathcal{B}}$ and $P_{\mathcal{B}^{\perp}}P_{\mathcal{B}^{\perp}} = P_{\mathcal{B}^{\perp}}$). It is easily shown that

$$\begin{aligned} -[B\times]^2 &= |B|^2 I - BB^T \\ &= |B|^2(I - bb^T) \\ &= |B|^2 P_{\mathcal{B}^{\perp}} \end{aligned} \tag{12}$$

The reaction wheels on a spacecraft produce torque in certain body directions. This forms the wheel control space $\mathcal{W}$. The directions along which no torque is produced by the wheels is its orthogonal complement $\mathcal{W}^{\perp}$.

The development below will be for a two-wheel system rather than a general wheel control space $\mathcal{W}$ to help facilitate understanding. For a two-wheel system that can produce torques along the y and z axes but not along the x-axis, we define the orthogonal projectors

$$P_x = xx^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad\qquad P_{yz} = I - xx^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

where $x = (1,\ 0,\ 0)^T$ is a unit vector along the x-axis. Then $T_w = P_{yz}$ and $P_{\mathcal{W}\perp} = P_x$. Note that $P_{x\perp} = I - P_x = P_{yz}$ and that $P_x P_{yz} = 0$.

The two wheels are mounted in the yz-plane at an angle θ from the body y-axis. The mapping T_w of wheel torques onto the body axes and the mapping $T_w^{\#}$ of a body torque command onto the wheel axes are given by

$$T_w = \begin{bmatrix} 0 & 0 \\ c & c \\ s & -s \end{bmatrix} \qquad\qquad T_w^{\#} = \begin{bmatrix} 0 & 1/2c & 1/2s \\ 0 & 1/2c & -1/2s \end{bmatrix} \tag{14}$$

where $c = \cos\theta$ and $s = \sin\theta$. Note that these mappings are not projectors.

Attitude Control Torque Distribution

The feedback torque for controlling attitude is computed from attitude and rate information using the PID algorithm in appendix C, although in principle any appropriate compensator design may be used. The desired attitude control torque $\tau_c \in \boldsymbol{R}^3$ comprises three components: a torque $\tau_{cw} \in \mathcal{W}$ that can be generated by the wheels, a torque $\tau_{cm} \in \mathcal{B}^{\perp}$ that can be generated by magnetic torque bars or coils, and a torque $\tau_{ce} \in \boldsymbol{R}^3$ that cannot be generated at all by the actuators. This is written

$$\tau_c = \tau_{cw} + \tau_{cm} + \tau_{ce}\ . \tag{15}$$

Note that τ_{ce} can be zero only $\mathcal{B}^{\perp}$ is orthogonal to $\mathcal{W}$ or $\tau_c \in \mathcal{W}$ or $\tau_c \in \mathcal{B}^{\perp}$. Expressions are now given for these torques in terms of τ_c by using and subspace projections.

Since $\tau_{cm} \in \mathcal{B}^{\perp}$, the remaining torque is in $\mathcal{B}$ and projected onto $\mathcal{W} = P_{yz}$ and $\mathcal{W}^{\perp} = P_x$. Then

$$\tau_{cw} = P_{yz} P_{\mathcal{B}} \tau_c \qquad \text{and} \qquad \tau_{cm} = P_{\mathcal{B}\perp} \tau_c \qquad (\gamma = 0) \tag{16}$$

Because the available magnetic control torque is generally small compared to the wheel torque, it may be desirable to apportion some of the commanded magnetic torque that lies in $\mathcal{W}$ to the wheels and less to the magnetic torque bars or coils. Then

$$\tau_{cw} = P_{yz} P_{\mathcal{B}}\, \tau_c + \gamma P_{yz} P_{\mathcal{B}\perp} \tau_c \tag{17}$$

$$\tau_{cm} = P_{\mathcal{B}\perp} \tau_c - \gamma P_{\mathcal{B}\perp} P_{yz} \tau_c \tag{18}$$

where $0 \le \gamma \le 1$. Two extreme cases are $\gamma = 0$ (16) and $\gamma = 1$, for which we have

$$\tau_{cw} = P_{yz} \tau_c \qquad \text{and} \qquad \tau_{cm} = P_{\mathcal{B}\perp} P_x \tau_c \qquad (\gamma = 1) \tag{19}$$

Note that τ_{cw} is constructed to lie in $\mathcal{W}$ and τ_{cm} is constructed to lie in $\mathcal{B}^{\perp}$. The magnetic dipole moment that exactly achieves the total magnetic control torque τ_{cm} is computed from (53) in appendix B.

The part of the commanded attitude control torque τ_c that cannot be produced by either the wheels or magnetically is given by

$$\begin{aligned} \tau_{ce} &= \tau_c - \tau_{cw} - \tau_{cm} \\ &= (I - P_{yz} P_{\mathcal{B}} - \gamma P_{yz} P_{\mathcal{B}\perp} - P_{\mathcal{B}\perp} + \gamma P_{\mathcal{B}\perp} P_{yz}) \tau_c \\ &= (P_{\mathcal{B}} - P_{yz} P_{\mathcal{B}} + \gamma (P_{\mathcal{B}\perp} P_{yz} - P_{yz} P_{\mathcal{B}\perp})) \tau_c \\ &= (P_x P_{\mathcal{B}} + \gamma (P_{\mathcal{B}\perp} P_{yz} - P_{yz} P_{\mathcal{B}\perp})) \tau_c\, . \end{aligned} \tag{20}$$

This is simply the commanded torque in the direction of B that is also in the x-axis direction in body coordinates (not in the wheel torque space) and a portion of the commanded torque that cannot be produced simultaneously by the wheels and the torque rods. The term multiplying γ in (21) is related to the "momentum management" axis n as shown in appendix D. As shown in appendix D, we have $P_{\mathcal{B}^\perp} P_{yz} - P_{yz} P_{\mathcal{B}^\perp} = P_{\mathcal{B}} P_x - P_x P_{\mathcal{B}}$, so

$$\begin{aligned} \tau_{ce} &= (P_x P_{\mathcal{B}} + \gamma(P_{\mathcal{B}} P_x - P_x P_{\mathcal{B}}))\tau_c \\ &= (\gamma P_{\mathcal{B}} P_x + (1-\gamma) P_x P_{\mathcal{B}})\tau_c \,. \end{aligned} \tag{21}$$

For $\gamma = 0$ we have $\tau_{ce} = P_x P_{\mathcal{B}}$ and for $\gamma = 1$ we have $\tau_{ce} = P_{\mathcal{B}} P_x$.

Momentum Control

Let the momentum control torque in wheel coordinates be given by $\tau_u = T_w G_w (h_o - h_w)$ where G_w is a feedback compensator. Note that $\tau_u = P_{yz}\tau_u$ by the construction of T_w. Projecting τ_u onto the magnetic control subspace we have

$$\tau_h = P_{\mathcal{B}^\perp} T_w G_w (h_o - h_w) \,. \tag{22}$$

The total magnetic control torque, which includes wheel momentum control torque, is

$$\tau_m = \tau_{cm} + \tau_h \,. \tag{23}$$

The wheels can only produce torque in $\mathcal{R}(P_{yz})$, so the total wheel torque on the spacecraft is

$$\tau_w = P_{yz}\tau_h - \tau_{cw} \,. \tag{24}$$

The feedback to the wheels is then given by

$$\tau_{wf} = T_w^{\#}\tau_w \,. \tag{25}$$

The wheel momentum control torque causes a disturbance torque given by

$$\begin{aligned} \tau_{he} &= \tau_h - P_{yz}\tau_h \\ &= P_x \tau_h \\ &= P_x P_{\mathcal{B}^\perp}\tau_u \,. \end{aligned} \tag{26}$$

Although $\tau_h \in \mathcal{B}^\perp$ can be produced without error, the generated wheel unloading torque $P_{yz}\tau_h \notin \mathcal{B}^\perp$, so the magnetic momentum control creates the disturbance τ_{he}. Alternatively we could set $\tau_h = P_{yz}\tau_u$ so that $\tau_w = \tau_h - P_{yz}\tau_{cw}$, but then $\tau_{he} = \tau_h - P_{\mathcal{B}^\perp}\tau_h = P_{\mathcal{B}}\tau_u = P_{\mathcal{B}} P_{yz}\tau_u$. See remark 4 below.

The Direct Subspace Projection method is illustrated in the block diagram in Figure 1. This block diagram implements (17), (18), and (23)–(25). The wheel geometry and axis projections shown are for the two-wheel system used in the simulation results in the "RESULTS" section, but in general can be for any system configuration.

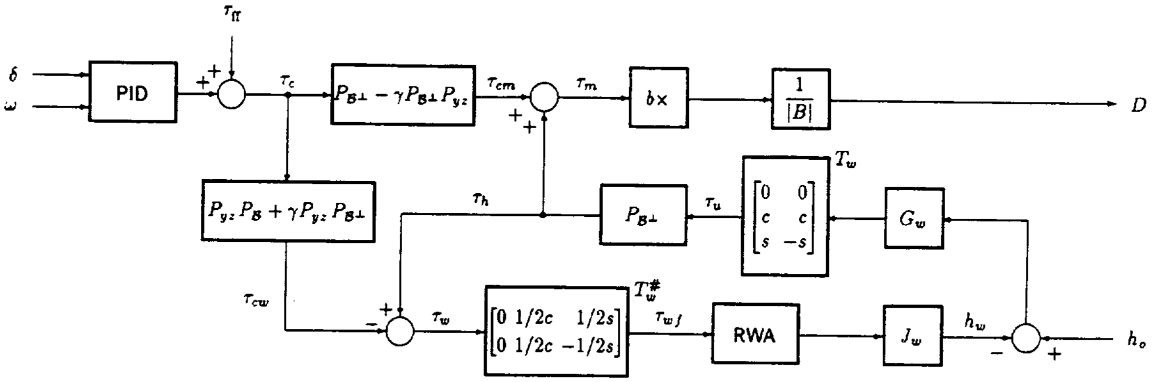


Figure 1: Block diagram of the Direct Subspace Projection algorithm

Remarks

1. The projection matrices do not have to be formed directly, thus saving considerable memory and computation. For example, $\hat{\tau} = P_{\mathcal{B}}\tau = bb^T\tau = b(b^T\tau)$, which is simply a vector times a scalar. Then $P_{\mathcal{B}\perp} = (I - bb^T)\tau = \tau - \hat{\tau}$, which is simply a vector subtraction. Projections using P_{yz} and P_x are simply a matter of extracting the appropriate components of a vector.
2. In general, γ can be a function of frequency. If it is a high-pass function, then only cyclical torques will go to the wheels and constant torques will be sent to the magnetic torquers. Frequency weighted torque distribution is discussed in ref. 2.
3. Although conceptually any torque vector should be producible when B does not lie in $\mathcal{W}^{\perp}$, the torque error equation (21) shows that the desired torque cannot be produced exactly. This is a consequence of minimum-norm projections. However, γ may be computed from (21) to minimize this term with the restriction $0 \leq \gamma \leq 1$. The method on COMET discussed previously attempts to produce the commanded torque, but that causes actuator saturation whenever the magnetic field is near the axis with no wheel torque.
1
4. One could consider projecting the momentum control torque τ_u onto the "momentum management" axis n defined in (3) so that $\tau_h = P_n\tau_u$. This would make τ_{he} zero since then $\tau_u = P_{yz}\tau_u$ and $\tau_u = P_{\mathcal{B}\perp}\tau_u$, i.e., $\tau_u \in \mathcal{W} \cap \mathcal{B}^{\perp}$. The trade off is that less momentum control is available. Also the n-axis is not well-defined when B is nearly aligned with x, but that is the condition when the maximum momentum control torque is available. The relationship between the projector P_n and the projectors P_{yz} and $P_{\mathcal{B}\perp}$ is derived in appendix D.

RELATIONSHIP BETWEEN METHODS

It is shown in this section how the Direct Subspace Projection method is related to the indirect methods described in the "TORQUE DISTRIBUTION METHODS" section.

Relationship Between the Direct Method and the CLARK Method

Equation (1) can be solved directly without using the QR decomposition. From (1) then,

$$\begin{bmatrix} I & 0 & 0 \\ [B\times] & [B\times] & B \end{bmatrix} \begin{bmatrix} I & -[B\times] \\ 0 & -[B\times] \\ 0 & B^T \end{bmatrix} \begin{bmatrix} \tau_w \\ D \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ [B\times] & [B\times] & B \end{bmatrix} \begin{bmatrix} \tau_{att} \\ \tau_h \\ 0 \end{bmatrix} \tag{27}$$

from which we get

$$\begin{bmatrix} I & -[B\times] \\ [B\times] & BB^T - 2[B\times]^2 \end{bmatrix} \begin{bmatrix} \tau_w \\ D \end{bmatrix} = \begin{bmatrix} \tau_{att} \\ [B\times](\tau_{att} + \tau_h) \end{bmatrix} \tag{28}$$

The inverse of the matrix on the left is easily computed analytically, and so

$$\begin{bmatrix} \tau_w \\ D \end{bmatrix} = \frac{1}{|B|^2} \begin{bmatrix} 2|B|^2 I - BB^T & [B\times] \\ -[B\times] & I \end{bmatrix} \begin{bmatrix} \tau_{att} \\ [B\times](\tau_{att} + \tau_h) \end{bmatrix} \tag{29}$$

Finally we get the common result

$$\begin{aligned} \begin{bmatrix} \tau_w \\ D \end{bmatrix} &= \begin{bmatrix} \tau_{att} + \frac{1}{|B|^2}[B\times]\tau_h \\ \frac{1}{|B|^2}[B\times]\tau_h \end{bmatrix} \\ &= \begin{bmatrix} \tau_{att} + [b\times]^2\tau_h \\ \frac{1}{|B|}[b\times]\tau_h \end{bmatrix} . \end{aligned} \tag{30}$$

This can be written

$$D = \frac{1}{|B|}[b\times]\tau_h \tag{31}$$

$$\tau_h' = D \times B \tag{32}$$

$$\tau_w = \tau_{att} - \tau_h' \tag{33}$$

Evidently this result could have been obtained directly from (1) by solving for D using the second and third rows of the matrix equation and then substituting the result into the first.

Except for the definition of τ_w, which differs only in sign, this is the same result obtained for the Direct Subspace Projection approach with $\gamma = 1$, i.e., none of the attitude control torque τ_{att} is generated magnetically. The Direct Subspace Projection method has the advantage that part of the attitude control torque is generated magnetically for $\gamma < 1$.

The relationship between the Direct Subspace Projection method and the method used on CLARK is now shown for the case where the roll wheel is eliminated. Equation (2) can be written more succinctly, with some reordering of the rows and setting $\tau_{hx} = 0$, as

$$\begin{bmatrix} P_{yz} & -[B\times] \\ 0 & -P_{yz}[B\times] \\ x^T & 0 \\ 0 & B^T \end{bmatrix} \begin{bmatrix} \tau_w \\ D \end{bmatrix} = \begin{bmatrix} \tau_{att} \\ \tau_h \\ 0 \\ 0 \end{bmatrix} \tag{34}$$

The solution is easily computed analytically by following the same procedure as before, and so

$$\begin{bmatrix} \tau_w \\ D \end{bmatrix} = \frac{1}{|B|^2} \begin{bmatrix} |B|^2 I - P_{yz}[B\times]^2 P_{yz} & P_{yz}[B\times] \\ -[B\times]P_{yz} & I \end{bmatrix} \begin{bmatrix} P_{yz}\tau_{att} \\ [B\times](\tau_{att} + P_{yz}\tau_h) \end{bmatrix} \tag{35}$$

With a few manipulations we get

$$\tau_w = P_{yz}\tau_{att} + P_{yz}[B\times]^2(P_x\tau_{att} + P_{yz}\tau_h) \tag{36}$$

$$D = \frac{1}{|B|^2}[B\times](P_x\tau_{att} + P_{yz}\tau_h) \tag{37}$$

This can be written

$$\tau_h' = D \times B \tag{38}$$

$$\tau_w = P_{yz}\tau_{att} - P_{yz}\tau_h' \tag{39}$$

This is the same result obtained for the Direct Subspace Projection method with $\gamma = 1$.

This again establishes the relationship between the CLARK algorithm and the Direct Subspace Projection method. However, the Direct Subspace Projection method requires very little computation and has a clear geometrical interpretation. Although the CLARK algorithm was said to require that a set of equations be formulated and solved by using QR decomposition and backward substitution (ref. 5), it was shown here that it can be solved in closed-form and that the solution is equivalent to the Direct Subspace Projection method with $\gamma = 1$. The Direct Subspace Projection method is also more general because some of the magnetic torque that lies in the wheel space can be apportioned to the wheels and less to the magnetic torque bars or coils for $\gamma < 1$.

Relationship Between the Direct Method and the OrbComm Method

The formulation of the OrbComm torque distribution algorithm in equations (9) and (10) unfortunately does not result in a simple closed-form solution and cannot be related directly to the Direct Subspace Projection method as was the CLARK formulation, although there are similarities.

Applying the Direct Subspace Projection method to the OrbComm system with $T_w = P_z$ and $P_{\mathcal{W}\perp} = P_{xy}$ yields the simple equations

$$\tau_{cw} = P_z P_{\mathcal{B}}\,\tau_c + \gamma P_z P_{\mathcal{B}\perp}\tau_c \tag{40}$$

$$\tau_{cm} = P_{\mathcal{B}\perp}\tau_c - \gamma P_{\mathcal{B}\perp} P_z \tau_c\,, \tag{41}$$

where

$$P_z = zz^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad P_{xy} = I - zz^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{42}$$

and where $z = (0,\ 0,\ 1)^T$ is a unit vector along the z-axis. For the two extreme cases, $\gamma = 0$ and $\gamma = 1$, we have

$$\tau_{cw} = P_z P_{\mathcal{B}}\,\tau_c \qquad \tau_{cm} = P_{\mathcal{B}\perp}\tau_c \qquad (\gamma = 0) \tag{43}$$

$$\tau_{cw} = P_z\tau_c \qquad \tau_{cm} = P_{\mathcal{B}\perp}P_{xy}\tau_c \qquad (\gamma = 1) \tag{44}$$

For any $0 \leq \gamma \leq 1$ the command error torque is given by

$$\tau_{ce} = (\gamma P_{\mathcal{B}} P_{xy} + (1-\gamma) P_{xy} P_{\mathcal{B}}) \tau_c \, . \tag{45}$$

Other quantities are

the total magnetic control torque	$\tau_m = \tau_{cm} + \tau_h$	(46)
the wheel momentum control torque	$\tau_h = -P_{\mathcal{B}\perp} P_z G_w h_w$	(47)
the total torque on the wheel	$\tau_w = P_z \tau_h - \tau_{cw}$	(48)
the feedback torque	$\tau_{wf} = z^T \tau_w$	(49)
and the momentum control error torque	$\tau_{he} = P_{xy} \tau_h \, .$	(50)

RESULTS

Simulation results are shown in Figure 2 for $\gamma = 0$ and $\gamma = 1$. Although most parameters are taken from TSX-5, enviromental torques are modeled as $0.0001(1+ \sim \omega_o t + \cos \omega_o t)$ in pitch rather than as a small constant. The pitch loop uses PID compensation and the roll/yaw control is PD compensated. The initial attitude was offset from the target orbital frame by 5 degrees in roll, pitch, and yaw. The initial wheel momenta were -1.5 and -1.4 N·m·s for the a and b wheels. These initial conditions were chosen to create a transient condition for the purpose of illustrating the torque distribution algorithm. As can be seen by comparing the results for $\gamma = 0$ and $\gamma = 1$, the most notable difference is in the transient response. Clearly more magnetic dipole is commanded when $\gamma = 0$. What is not clear because of the scale of the graph of wheel torque is that the initial wheel torques peaked at 20 mN·m. Slightly less disturbance is seen in the attitude error for $\gamma = 0$ and the disturbance is at a lower frequency. This is a result of modulation that occurs when $\gamma = 1$.

CONCLUSION

A new torque distribution algorithm was developed in this paper that has a solid mathematical foundation, requires little computation, and does not exhibit singularities or other undesirable behavior. It can be analyzed because it has a simple closed-form solution and a clear geometric interpretation. This algorithm is denoted the Direct Subspace Projection or DSP method.

Torque distribution algorithms that were developed for and implemented on the CLARK, COMET, and OrbComm satellites were reviewed in this paper and compared with the DSD method. It was shown that the DSD method solves the CLARK torque distribution equations but the new algorithm requires far less computation. The CLARK, COMET, and OrbComm algorithms were all implemented by solving a set of equations by QR decomposition. The COMET and OrbComm algorithms suffer from a singularity (ill-conditioning), although the OrbComm algorithm was modified to alleviate that problem. Equations using the Direct Subspace Projection method for CLARK, COMET, and OrbComm were given in this paper. A closed-form solution for the CLARK algorithm was given in this paper and was shown to be the same as the Direct Subspace Projection solution with $\gamma = 1$. The Direct Subspace Projection method is illustrated with simulation results designed to demonstrate the effect of the subspace projections on actuator effort.

REFERENCES

1. Mark Pittelkau, *Optimal Periodic Control for Spacecraft Pointing and Attitude Determination*, AIAA J. Guidance, Control, and Dynamics, Vol. 16, No. 6, Nov–Dec 1993, pp. 1078–1084.

2. Mark Pittelkau, *Frequency Weighted LQG Control of Spacecraft Attitude*, 1st IEEE Conference on Control Applications, Vol. 1, IEEE, New York, Sept. 1992, pp.336–341.

3. Xiao-jiang Chen and Willem Steyn, *Optimal Combined Reaction-Wheel Momentum Management for LEO Earth-Pointing Satellites*, 12th AIAA/USU Conference on Small Satellites, Paper No. SSC98-IX-2, 1998.

4. *Attitude Determination and Control Subsystem DDC*, CTA Space Systems, 5 April 1995.

5. Walter Grossman and Douglas Freesland, *The Control of Satellites with Microgravity Constraints: The Comet Control System*, NASA/GSFC Flight Mechanics and Estimation Symopsium, 17–19 May 1994.

6. Mark Krebs, *A New Attitude Control Mechanism For LEO Satellites*, 10th AIAA/Utah State U. Conference on Small Satellites, Sept 16–19, 1996.

7. Paul M. Stolz, Mark Krebs, and Rick Baltman, *OrbComm Attitude Determination and Control*, AIAA Paper No. 96-3620.

8. E. Kreyszig, *Introductory Functional Analysis With Applications*, New York: John Wiley and Sons, 1978.

APPENDICES

A NOMENCLATURE AND NOTATION

B	geomagnetic field vector
b	unit geomagnetic field vector
τ_w	total wheel control torque (24)
τ_m	total magnetic control torque (23)
τ_h	momentum control torque (22)
τ_c	attitude control torque (15) ($\equiv \tau_{att}$ in (1))
τ_w	total wheel control torque (24)
τ_m	total magnetic control torque (23)
τ_{wf}	momentum control feedback to the wheels (25)
τ_{cw}	attitude control torque to the wheels (17)
τ_{cm}	attitude control torque to the torque rods (18)
τ_{ce}	attitude control error torque (15), (21)
τ_{he}	momentum control error torque (26)
τ_{gg}	gravity gradient torque
τ_{gy}	gyroscopic torque
τ_d	environmental disturbance torque
$\mathcal{B}^{\perp}$	subspace containing all possible magnetic control torques
$\mathcal{W}$	subspace containing all possible wheel torques
I	identity matrix

$P_{\mathcal{B}}$	projector onto $\mathcal{B}$ (11)
$P_{\mathcal{B}^\perp}$	projector onto $\mathcal{B}^\perp$ (11)
P_{yz}	projector onto $\mathcal{W}$ (13)
P_x	projector onto $\mathcal{W}^\perp$ (13)
T_w	mapping of wheel torque onto body (14)
$T_w^\#$	mapping of body torque onto wheel axes (14)
$\boldsymbol{R}^3$	3-space
$\mathcal{R}(P)$	range space of a projector P
$[u\times]$	cross product matrix $[u\times]v = u \times v$
D	magnetic torquer dipole moment (53)
H	total momentum (54)
ω	body angular rate in body coordinates (54)
δ	attitude error (57)
J	spacecraft inertia matrix (57)

B DIPOLE MOMENT COMPUTATION

The control law (23) computes a commanded magnetic torque τ_{cm} that is in $\mathcal{R}(P_{\mathcal{B}^\perp})$, which is the plane perpendicular to B, so that $\tau_{cm} = P_{\mathcal{B}^\perp}\tau_{cm}$. Thus the commanded dipole D is also perpendicular to B and in the direction of $B \times \tau_{cm}$. The dipole must result in a torque equal to τ_{cm}, so $D \times B = \tau_{cm}$. Since $D = kB \times \tau_{cm}$, where k is a scalar, we have

$$\begin{aligned} D \times B &= k(B \times \tau_{cm}) \times B \\ &= -kB \times (B \times \tau_{cm}) \\ &= -k[B\times]^2\tau_{cm} \\ &= k|B|^2(I - bb^T)\tau_{cm} \\ &= k|B|^2 P_{\mathcal{B}^\perp}\tau_{cm} \\ &= k|B|^2\tau_{cm} \end{aligned} \tag{51}$$

and so

$$k = \frac{1}{|B|^2} \,. \tag{52}$$

The magnetic dipole command that produces the torque τ_{cm} is then

$$\begin{aligned} D &= \frac{1}{|B|^2} B \times \tau_{cm} \\ &= \frac{1}{|B|} b \times \tau_{cm} \end{aligned} \tag{53}$$

C ATTITUDE CONTROLLER

The dynamics of a spacecraft are governed by Euler's equation

$$\dot{H} + \omega \times H = \tau_{cm} + \tau_{gg} + \tau_d \tag{54}$$

Since the total momentum is $H = J\omega + h_w$, the gyroscopic torque is $\tau_{gy} = \omega \times H$, and the wheel torque is given by $\dot{h}_w = \tau_w$,

$$J\dot{\omega} + \tau_w + \tau_{gy} = \tau_{cm} + \tau_{gg} + \tau_d \tag{55}$$

The wheel torque comprises two components: a wheel momentum control torque τ_h and an attitude control torque τ_{cw} from the wheel acting on the body. Then $\tau_w = \tau_h - \tau_{cw}$. The equation of motion is now

$$J\dot{\omega} + \tau_h + \tau_{gy} = \tau_{cw} + \tau_{cm} + \tau_{gg} + \tau_d \tag{56}$$

Suppose the attitude control torque is given by the PID control law

$$\tau_c = -J\left[K_p\delta + K_r(\omega - A\omega_o) + K_i \int \delta\, dt\right] + \tau_{gy} - \tau_{gg} \tag{57}$$

D RELATIONSHIP BETWEEN THE PROJECTOR P_n AND THE PROJECTORS P_{yz} AND $P_{\mathcal{B}\perp}$

The relationship between the projector P_n and the projectors P_{yz} and $P_{\mathcal{B}\perp}$ is derived in this section. Let the subspace $\mathcal{B}^\perp$ contain all possible magnetic torques and the space $\mathcal{W}$ contain all possible wheel torques. Suppose that $v \in \boldsymbol{N}$, i.e., v lies along the intersection of the space $\mathcal{B}^\perp$ and the space $\mathcal{W}$. Let $\mathcal{W} = P_{yz}$ for illustration. Then we have $v = P_{\mathcal{B}\perp}v$, $v = P_{yz}v$, and $v = P_n v$. Substituting the first two of these relations into each other we get $v = P_{yz}P_{\mathcal{B}\perp}v$ and $v = P_{\mathcal{B}\perp}P_{yz}v$. Thus we conclude that $P_{yz}P_{\mathcal{B}\perp}v = P_{\mathcal{B}\perp}P_{yz}v$ and so

$$(P_{yz}P_{\mathcal{B}\perp} - P_{\mathcal{B}\perp}P_{yz})v = 0\,. \tag{58}$$

This can be written

$$((I - P_x)(I - P_{\mathcal{B}}) - (I - P_{\mathcal{B}})(I - P_x))v = 0 \tag{59}$$

and so

$$(P_xP_{\mathcal{B}} - P_{\mathcal{B}}P_x)v = 0\,. \tag{60}$$

Multiply by the transpose of the skew-symmetric matrix on the left and by a scalar k^2 to get

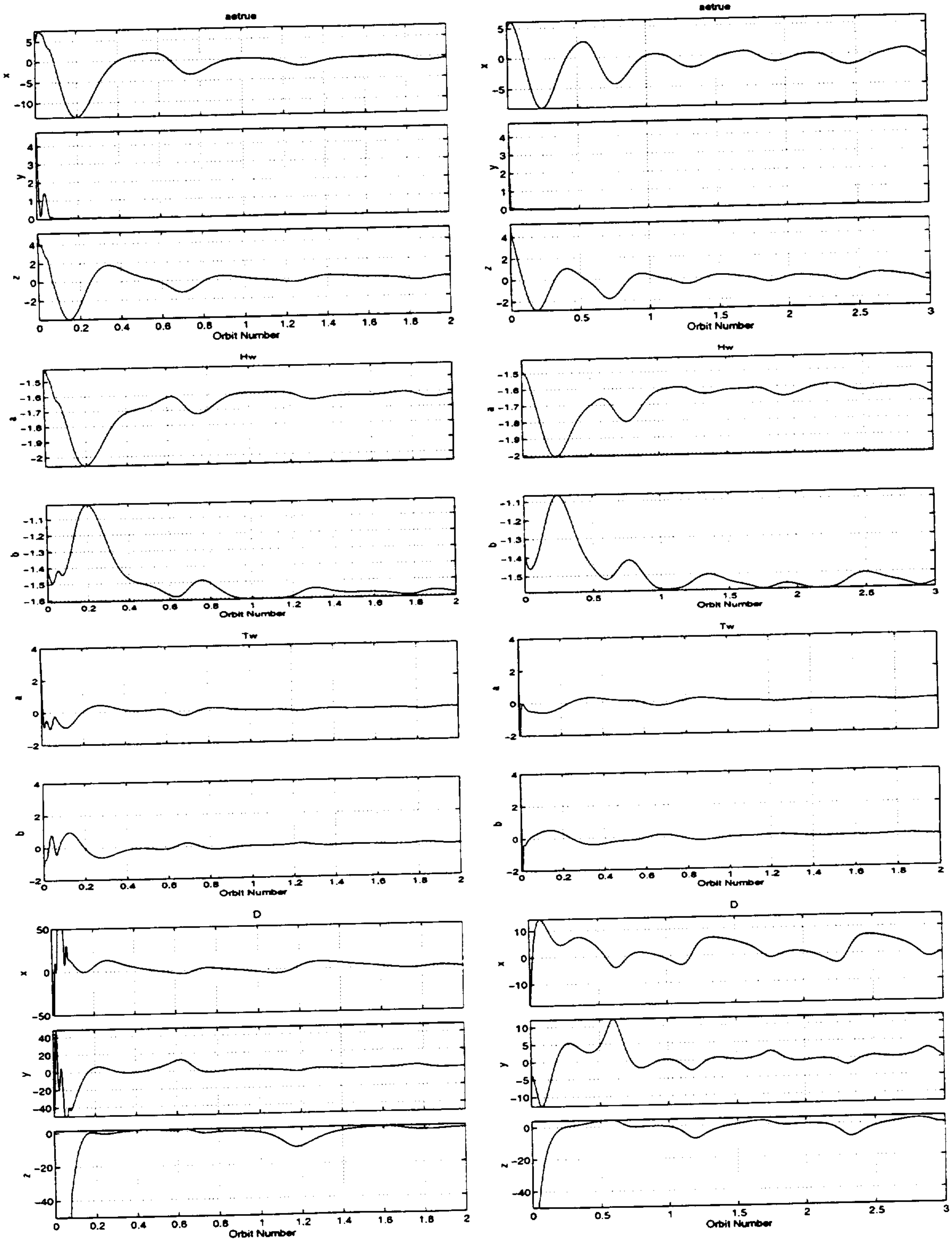
$$-k^2(P_xP_{\mathcal{B}} - P_{\mathcal{B}}P_x)^2v = 0\,. \tag{61}$$

Recall that $v \in \boldsymbol{N}$. We have $P_n = nn^T$ and $P_{n\perp} = I - nn^T = -[n\times]^2$. It is now shown that $P_{n\perp} = -k(P_xP_{\mathcal{B}} - P_{\mathcal{B}}P_x)^2$.

Since $P_{\mathcal{B}} = bb^T$ and $P_x = xx^T$ we have

$$\begin{aligned} P_xP_{\mathcal{B}} - P_{\mathcal{B}}P_x &= xx^Tbb^T - bb^Txx^T \\ &= (b^Tx)(xb^T - bx^T) \\ &= (b^Tx)[(b \times x)\times] \end{aligned} \tag{62}$$

Letting $k = \dfrac{1}{(b^Tx)|b \times x|}$ and defining $n = \dfrac{b \times x}{|b \times x|}$ we conclude that $[n\times] = k(P_xP_{\mathcal{B}} - P_{\mathcal{B}}P_x)$. Therefore $P_{n\perp} = -[n\times]^2 = -k^2(P_xP_{\mathcal{B}} - P_{\mathcal{B}}P_x)^2$ and $P_n = nn^T$.



$\gamma = 0$ $\gamma = 1$

Figure 2: Attitude error (aetrue) in deg, wheel momentum (Hw) in N·m·s, wheel torque (Tw) in mN·m, magnetic dipole (D) in A·m^2 for $\gamma = 0$ (left column) and $\gamma = 1$ (right column).

IN-FLIGHT STATIC AND DYNAMIC IMBALANCE ESTIMATION AND CONTROL OF A DUAL-SPIN SATELLITE

Walter D. Grossman
Applied Systems Engineering, Inc.
wgrossman@ase.wl.com

ABSTRACT

A radar satellite is equipped with a large payload spinning along the yaw axis at 30 RPM. The payload is equipped with a counter-rotating momentum wheel to null the yaw component of the momentum. Though the payload is statically and dynamically balanced prior to integration it is anticipated that the mass properties will change in flight due to out-gassing and moisture evaporation. The change in the payload center-of-mass and products-of-inertia results in vehicle nutation and subsequent degradation of the radar images.

To provide for in-flight balancing the payload is equipped with two pairs of orthogonal control mass/lead screw devices mounted in two planes. An inertial measurement unit and an accelerometer package provide rotational and translational measurements. Due to mechanical reliability concerns, payload/satellite interface force sensors are not used in this application.

In this paper a novel technique is introduced to estimate the imbalance and to determine the compensating control mass motion. The estimator estimates the location of "virtual" control mass locations which yield the observed dynamic behavior. The actual control masses are moved to the axial mirror image positions to compensate for the mass imbalance. The actual mass properties are not estimated. Due to uncertainties in the nominal spacecraft and payload, the process is iterated two-three times until the nutation is eliminated.

The estimator uses a variation of a discrete-time nonlinear observer recently developed by this author. This observer is applicable to nonlinear discrete-time, full-state feedback systems in which the dynamics are affine in the unknown parameters. In the observer used here a "quasi-optimal" gain is derived from propagation of a Riccati-like discrete-time variance equation.

PROBLEM DESCRIPTION

A radar satellite is equipped with a large payload spinning along the yaw axis at 30 RPM. The payload is equipped with a counter-rotating momentum wheel to null the yaw component of the momentum. Though the payload is statically and dynamically balanced prior to integration it is anticipated that the mass properties will change in flight due to out-gassing and moisture evaporation. This change in the payload center-of-mass and products-of-inertia results in vehicle nutation and subsequent degradation of the radar images.

To provide for in-flight balancing, the payload is equipped with two pairs of orthogonal control mass/lead screw devices mounted in two planes. An inertial measurement unit and an accelerometer package provide rotational and translational measurements. Due to mechanical reliability concerns, payload/satellite interface force sensors, used on similar mission, are not used in this application. Figure 1 illustrates the system configuration.

The control mass/lead screw configuration with the four masses provides sufficient degrees of freedom to simultaneously remove the products-of-inertia terms and to move the center of mass of the spinning payload onto the spin axis. There are four quantities to null, viz., J_{xz}, J_{yz}, CM_x, and CM_y, and four independent control parameters, viz., the position of the control masses on their lead screws.

APPROACHES

The differential equation for the dynamics of the dual-spin satellite with a counter-rotating momentum wheel and four control masses is complicated and nonlinear. The dynamics generated using the SD-FAST modeling program. The results of this program was a Fortran subroutine (with many supporting routines) which propagated a twenty-nine element state vector form time t_1 to time t_2. An additional subroutine yielded the acceleration of a specified point on the vehicle.

The presence of some states of this large twenty-nine element state vector are a consequence of the required form that model data are presented to the dynamics code generation software and are not of interest.[1] The estimator and the control system were developed using a reduced-dimension sixteen element state vector and two four-element parameter vectors (twenty-four states). The actual execution of the integration used the entire state vector as required by the modeling software. This dynamic portion of the state vector is defined:

$$x_{\text{eom}} = \begin{bmatrix} q \\ \omega \\ R \\ V \\ \theta_{\text{pyld}} \\ \omega_{\text{pyld}} \\ \omega_{\text{whl}} \end{bmatrix} . \tag{1}$$

States q and ω represent, respectvely, the vehicle attitude quaternion and body rate. Vectors R and V represent, respectively, the position and velocity of the system geometric center relative to the system CM. States θ_{pyld} and ω_{pyld} represent, respectively, the payload angular position and velocity relative to the spacecraft. State ω_{whl} represents the speed of the counter-rotating momentum wheel.

The two parameter vectors p and μ are defined. Parameter vector p represents the locations of the control masses and parameter vector μ represents the four mass imbalance quantities of the payload, (excluding the mass distribution effects of the control masses), i.e.,

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \quad \mu = \begin{bmatrix} CM_x \\ CM_y \\ J_{xz} \\ J_{yz} \end{bmatrix} . \tag{2}$$

[1] These large state vectors appear to be a common consequence of automated dynamics software. One example of an uninteresting state is the momentum wheel position. It is required for the correct execution of the subroutines modeling the differential equation of motion, but it is not particularly interesting by itself. These state variables have the ill-effect of greatly increasing computational costs by extending the state vector and by driving downward the integration step size. Effectively, they significantly stiffen the system.

In this work, the spacecraft is uncontrolled with no torques on the reaction wheels. Torques are applied to the payload spin motor and the counter-spinning momentum wheel:

$$u = \begin{bmatrix} \tau_{\text{pyld}} \\ \tau_w \end{bmatrix}. \tag{3}$$

The payload and wheel control torques and the control mass positions are held constant over a sample period. Mass imbalance vector μ is presumed fixed. The resulting system dynamics are given by:

$$\dot{x} \;=\; \bar{f}[x(t), u(kT), p(kT), \mu].$$

The modeling software output the system state at discrete times

$$\begin{aligned} x_{k+1} &= x_k + \int_{kT}^{(k+1)T} \bar{f}[x(t), u(kT), p(kT), \mu]dt & (4) \\ &= f(x_k, u_k, p_k, \mu) & (5) \end{aligned}$$

To facilitate the development of the estimator, it is convienient to decompose the state vector x of (1)

$$x \;=\; \begin{bmatrix} x_a \\ x_b \end{bmatrix} \tag{6}$$

where

$$\begin{aligned} x_a &= \begin{bmatrix} q \\ R \\ V \\ \theta_{\text{pyld}} \\ \omega_{\text{pyld}} \\ \omega_{\text{whl}} \end{bmatrix} & (7) \\ x_b &= \omega. & (8) \end{aligned}$$

Using this partitioned state vector, the dynamics equation (5) is rewritten

$$\begin{bmatrix} x_a \\ x_b \end{bmatrix} = \begin{bmatrix} f_a(x_{ak}, x_{bk}, u_k, p_k, \mu) \\ f_b(x_{ak}, x_{bk}, u_k, p_k, \mu) \end{bmatrix}. \tag{9}$$

The spacecraft is equipped with a star camera, an Inertial Measurement Unit (IMU), and a three-axis accelerometer package. The acceleration of the vehicle at the location of the accelerometer package is not a state variable but rather is a complicated function of the state x, the control mass location p, the mass imbalance vector μ, the control u, and the fixed location of the accelerometer package. Using integration of the IMU output to provide the (short-term) estimates of attitude, and integration of the accelerometer to provide (short-term) estimates of the motion of the spacecraft geometric center, the output mapping equation is

$$y_a = x_a = \begin{bmatrix} q \\ R \\ V \\ \theta_{\text{pyld}} \\ \omega_{\text{pyld}} \\ \omega_{\text{whl}} \end{bmatrix} \tag{10a}$$

$$y_b = x_b = \omega \tag{10b}$$

$$y_c = \quad h_c(x_{ak}, x_{bk}, u_k, p_k, \mu), \quad \leftarrow \text{acceleration output equation} \tag{10c}$$

where an auxiliary measurement equation h_c has been defined for the acceleration output.

The initial approach to this problem was to apply the *extended* separation principle [ref. 1] in which we implement an observer to estimate μ and then the move the control masses to null them. The system dynamics are nonlinear in continuous time. The output map is nonlinear and sampled at discrete times. Control is applied at discrete times. The observer problem is to develop an algorithm which works for this nonlinear discrete-time system. The control problem is to determine how to move the masses given the estimates from the observer.

General Estimation Methods

Several estimation methods were reviewed for this application:

1. A twenty-four state extended Kalman filter.

2. A reduced-order extended Kalman filter.

3. A Grossman/Friedland nonlinear, discrete-time parameter estimator [ref. 2, 3, 4].

The most common approach is to augment the system state vector x with p and μ and apply an Extended Kalman Filter (EKF). For several reasons this approach was not attempted. Difficulties in the gain calculation was the primary reason for not implementing the EKF. The augmented state vector has twenty-four elements and the EKF requires the propagation of a 24 × 24 covariance matrix. This propagation would also require a state transition matrix which is not known and would have to be estimated by discrete differences. This propagation would require the calculation of the 24 × 24 discrete-difference partial derivative approximations of $\partial f_a/\partial x$ and $\partial f_b/\partial u$ (augmenting the proper state vector with p and mu yields twenty-for states) and 24 × 6 discrete-difference approximations of partial derivatives $\partial f_a/\partial u$ and $\partial f_b/\partial u$. Referring to output equations (10a)–(10c), estimation of the observation partials requires calculation of another 3×24 discrete-difference approximations of partial derivatives $\partial h_c/\partial x$. (The dynamic quantities other than the acceleration are directly measured, leading to unity partial derivatives.)

Since all the dynamic state variables are directly measured with high-accuracy, low-noise sensors, the reduced-order extended Kalman filter might have reduced the algorithm size and complexity which burdened the implementation of the full-state extended Kalman filter. The difficulty with the reduced-order (EKF) is that its derivation requires explicit knowledge of the state propagation equation and measurement mapping equation. Lacking this explicit knowledge, the filter would require calculation of discrete-difference approximations to partial derivatives. Its derivation was not entirely clear and in light of having to perform these discrete-differences, the benefit of using it over the full-state EKF was diminished.

Grossman/Friedland Parameter Estimator

In [ref. 4] Friedland introduced a parameter estimator for parameter-affine continuous-time systems in which the entire state was measured. These systems are of the form:

$$\begin{aligned} \dot{x} &= f(x,u)p \qquad (11) \\ y &= x. \qquad (12) \end{aligned}$$

In [ref. 3] Grossman extended the Friedland parameter estimator to discrete-time systems. For the discrete-time system

$$\begin{aligned} x_{k+1} &= f(x_k,u_k)p \qquad (13) \\ y_k &= x_k, \qquad (14) \end{aligned}$$

Grossman introduced the following parameter estimator

$$\begin{aligned} \hat{p}_k &= \hat{p}_{k-1} + K(y_{k-1})y_k + z_k \qquad (15) \\ z_{k+1} &= -K(y_k)f(x_k,u_k)\hat{p}_k, \qquad (16) \end{aligned}$$

which was stable if a gain function $K(y_k)$ could be found such that the matrix product $K(y_k)f(x_k,u_k)$ was symmetric with eigenvalues $|\lambda| < 1$.

In [ref. 2] Grossman developed three general expressions for $K(y_k)$, the choice of which depended upon the relative dimensions of x and p. For the situation where $\dim x > \dim p$, $K(y_k)$ can be the scaled left pseudoinverse. Suppressing the arguments of f for clarity, this expression for K is expressed:

$$K(y_k) = c(f'f)^{-1}f'. \qquad (17)$$

Grossman further developed [ref. 2] the quasi-optimal solution for $K(y_k)$ using the Riccati equation. Defining the measurement covariance W (noting that full state measurement is assumed), the initial parameter uncertainty covariance matrix P, and a covariance Q associated with dynamic process (13), the quasi-optimal parameter estimator is given:

$$\begin{aligned} \tilde{y}_{k+1} &= f(x_k,u_k)\hat{p}_k \qquad (18) \\ \tilde{P}_{k+1} &= \hat{P}_k \qquad (19) \\ K &= \tilde{P}_{k+1}f'(x_k,u_k)\left[W + f(x_k,u_k)\tilde{P}_{k+1}f'(x_k,u_k) + Q\right]^{-1} \qquad (20) \\ \hat{p}_{k+1} &= \hat{p}_k + K\left[y_{k+1} - \tilde{y}_{k+1}\right] \qquad (21) \\ \hat{P}_{k+1} &= \left[I - Kf(x_k,u_k)\right]\tilde{P}_{k+1}. \qquad (22) \end{aligned}$$

Note that in this situation where $p \in \mathcal{R}^4$, and $x \in \mathcal{R}^{24}$, the Grossman/Friedland estimator is far simpler to implement than the EKF.

APPLICATION OF THE GROSSMAN/FRIEDLAND ESTIMATOR

Method of Mass Imbalance Estimation

The quasi-optimal estimator was applied to this mass imbalance problem in an approximate fashion. Combining output equations (10a),(10b),(10c) with the propagation equations (9) yields

$$y_{ak+1} = f_a(x_{ak}, x_{bk}, u_k, p_k, \mu) \tag{23a}$$

$$y_{bk+1} = f_b(x_{ak}, x_{bk}, u_k, p_k, \mu) \tag{23b}$$

$$y_{ck+1} = f_c(x_{ak}, x_{bk}, u_k, p_k, \mu), \tag{23c}$$

where f_c is defined in terms of an acceleration output mapping function h_c and forward dynamics functions f_a and f_b:

$$f_c(x_{ak}, x_{bk}, u_k, p_k, \mu) \equiv h_c(y_{ak+1}, y_{bk+1}, u_k, p_k, \mu) \tag{24}$$

$$h_c(f_a(x_{ak}, x_{bk}, u_k, p_k, \mu), f_b(x_{ak}, x_{bk}, u_k, p_k, \mu), u_k, p_k, \mu). \tag{25}$$

Equations (23a), (23b), and (23c) are not in the parameter-affine form of the Grossman/Friedland estimator. By defining an output perturbation δy and a mass property perturbation $\delta\mu$, perturbation equation which are affine in the perturbed parameter μ can be written:

$$\delta y_{ak+1} = F_a^\mu(x_{ak}, x_{bk}, u_k, p_k, \mu)\delta\mu \tag{26a}$$

$$\delta y_{bk+1} = F_b^\mu(x_{ak}, x_{bk}, u_k, p_k, \mu)\delta\mu \tag{26b}$$

$$\delta y_{ck+1} = F_c^\mu(x_{ak}, x_{bk}, u_k, p_k, \mu)\delta\mu, \tag{26c}$$

where

$$F_a^\mu(x_{ak}, x_{bk}, u_k, p_k, \mu) = \frac{\partial f_a(x_{ak}, x_{bk}, u_k, p_k, \mu)}{\partial\mu} \tag{27}$$

$$F_b^\mu(x_{ak}, x_{bk}, u_k, p_k, \mu) = \frac{\partial f_b(x_{ak}, x_{bk}, u_k, p_k, \mu)}{\partial\mu} \tag{28}$$

$$F_c^\mu(x_{ak}, x_{bk}, u_k, p_k, \mu) = \frac{\partial f_c(x_{ak}, x_{bk}, u_k, p_k, \mu)}{\partial\mu} \tag{29}$$

The partial derivatives F_a^μ, F_b^μ, F_c^μ are not known explicitly since the underlying functions are determined by the numerical integration of (4). They can be estimated by numeric finite differences, but avoiding their calculation is the reason alternative methods to the EKF were sought after in the first place.

In the absence of high-frequency star camera attitude updates, "measurement" q of y_a is derived by integration of the IMU output. Likewise, quantities R and V are derived by integration of both the IMU output with the accelerometer output. Of all the quantities of y_a, all the "measurements," except for θ_{pyld}, ω_{pyld}, and ω_{whl}, are in fact derived by integration of sensor measurements. Of all the quantities in y_a, only θ_{pyld}, ω_{pyld}, and ω_{whl} represent new information. *Intuition* says that these measurements are not likely to be sensitive to mass imbalance. On the basis of this intuition, y_a was used in the propagation of the system dynamics, but was not used for parameter estimation updating. Eliminating y_a from parameter estimation updating saves the calculation of the 13×4 elements of F_a^μ.

Quantities F_b^μ and F_c^μ were estimated using central finite differences. Referring to (23b), (23c) and defining four elemental perturbation vectors $\delta\mu_i$:

$$\begin{aligned} \delta\mu_1 &\equiv \begin{bmatrix} \Delta & 0 & 0 & 0 \end{bmatrix}' && (30) \\ \delta\mu_2 &\equiv \begin{bmatrix} 0 & \Delta & 0 & 0 \end{bmatrix}' && (31) \\ \delta\mu_3 &\equiv \begin{bmatrix} 0 & 0 & \Delta & 0 \end{bmatrix}' && (32) \\ \delta\mu_4 &\equiv \begin{bmatrix} 0 & 0 & 0 & \Delta \end{bmatrix}', && (33) \end{aligned}$$

the partial derivatives of the ith rows of F_b^μ and F_c^μ are estimated:

$$\begin{aligned} {}^iF_b^\mu &\approx \frac{f_b(x_{ak}, x_{bk}, u_k, p_k, \mu + \delta\mu_i) - f_b(x_{ak}, x_{bk}, u_k, p_k, \mu - \delta\mu_i)}{2\Delta} && (34) \\ {}^iF_c^\mu &\approx \frac{f_c(x_{ak}, x_{bk}, u_k, p_k, \mu + \delta\mu_i) - f_c(x_{ak}, x_{bk}, u_k, p_k, \mu - \delta\mu_i)}{2\Delta}. && (35) \end{aligned}$$

The quasi-optimal mass properties estimator is thus summarized:

1. Initialize estimate of mass imbalance vector $\hat{\mu}$ (probably to zero) and 4×4 estimate covariance matrix P.

2. Propagate forward dynamics:

$$\begin{aligned} y_{ak+1} &= f_a(x_{ak}, x_{bk}, u_k, p_k, \hat{\mu}_k) \\ y_{bk+1} &= f_b(x_{ak}, x_{bk}, u_k, p_k, \hat{\mu}_k) \\ y_{ck+1} &= f_c(x_{ak}, x_{bk}, u_k, p_k, \hat{\mu}_k). \end{aligned}$$

3. Propagate perturbed dynamics for y_b, y_c

$$\begin{aligned} y_{bk+1}^{i+} &= f_b(x_{ak}, x_{bk}, u_k, p_k, \hat{\mu}_k + \delta\mu_i) \\ y_{ck+1}^{i-} &= f_b(x_{ak}, x_{bk}, u_k, p_k, \hat{\mu}_k - \delta\mu_i) \\ y_{bk+1}^{i+} &= f_c(x_{ak}, x_{bk}, u_k, p_k, \hat{\mu}_k + \delta\mu_i) \\ y_{ck+1}^{i-} &= f_c(x_{ak}, x_{bk}, u_k, p_k, \hat{\mu}_k - \delta\mu_i). \end{aligned}$$

4. Estimate partial derivatives F_b^μ and F_c^μ. The ith rows are given:

$$\begin{aligned} {}^iF_b^\mu &\approx \frac{y_{bk+1}^{i+} - y_{bk+1}^{i-}}{2\Delta} \\ {}^iF_c^\mu &\approx \frac{y_{bk+1}^{i+} - y_{ck+1}^{i-}}{2\Delta} \end{aligned}$$

5. Calculate Gain:

$$\begin{aligned}\tilde{P}_{k+1} &= \hat{P}_k \\ K &= \tilde{P}_{k+1}\begin{bmatrix} F_b^{\mu} \\ F_c^{\mu} \end{bmatrix}'\left[W + \begin{bmatrix} F_b^{\mu} \\ F_c^{\mu} \end{bmatrix}\tilde{P}_{k+1}\begin{bmatrix} F_b^{\mu} \\ F_c^{\mu} \end{bmatrix}' + Q\right]^{-1}\end{aligned}$$

6. Update estimate $\hat{\mu}$:

$$\hat{\mu}_{k+1} = \hat{\mu}_k + K\left\{\begin{bmatrix} y_{bk+1} \\ y_{ck+1} \end{bmatrix} - \begin{bmatrix} \hat{y}_{bk+1} \\ \hat{y}_{ck+1} \end{bmatrix}\right\} \tag{36}$$

7. Update covariance matrix P:

$$\hat{P}_{k+1} = \left\{I - K\begin{bmatrix} F_b^{\mu} \\ F_c^{\mu} \end{bmatrix}\right\}\tilde{P}_{k+1}$$

A typical simulation run is shown in figure 2.

Method of Virtual Masses

One problem with the method of the previous section is that even with the mass imbalances estimated, it is necessary to determine where to position the balance correction masses. The solution depends upon the system mass properties for which only four of them are estimated. Difficulties in developing a converging *control law* led to the development of the simpler method of "Virtual Masses."

The basis for the method of virtual masses is that a system with a mass *imbalance* and control weights parked on the spin axis and can conceptually be replaced with a *balanced* system in which "virtual" mass imbalance control weights are located in specific positions off the rotational axis of the payload. In other words, the control weights can be used to generate product terms and CM offsets of an otherwise balanced system. The method of virtual masses is to estimate the location of these "virtual" control masses in the assumed balanced system and command the true control weights to the positions that are axisymmetric opposite to the estimated virtual masses. Figure 3 illustrates the technique.

The estimator of the position of the virtual masses is conceptually very similar to the mass imbalance estimator described in the previous section. Th difference is that instead of estimating $\hat{\mu}$, the observer estimates a "virtual" $\hat{p}$, assuming $\mu = 0$. Setting $\mu = 0$, (23a)-(23c) are replaced by

$$\begin{aligned}y_{ak+1} &= f_a(x_{ak}, x_{bk}, u_k, p, 0) && \text{(37a)} \\ y_{bk+1} &= f_b(x_{ak}, x_{bk}, u_k, p, 0) && \text{(37b)} \\ y_{ck+1} &= f_c(x_{ak}, x_{bk}, u_k, p, 0), && \text{(37c)}\end{aligned}$$

from which (26a)-(26c) are replaced by

$$\begin{aligned}\delta y_{ak+1} &= F_a^p(x_{ak}, x_{bk}, u_k, p, 0)\delta p && \text{(37a)} \\ \delta y_{bk+1} &= F_b^p(x_{ak}, x_{bk}, u_k, p, 0)\delta p && \text{(37b)} \\ \delta y_{ck+1} &= F_c^p(x_{ak}, x_{bk}, u_k, p, 0)\delta p, && \text{(37c)}\end{aligned}$$

where

$$F_a^p(x_{ak}, x_{bk}, u_k, p, 0) = \frac{\partial f_a(x_{ak}, x_{bk}, u_k, p_k, 0)}{\partial p} \tag{38}$$

$$F_b^p(x_{ak}, x_{bk}, u_k, p, 0) = \frac{\partial f_b(x_{ak}, x_{bk}, u_k, p, 0)}{\partial p} \tag{39}$$

$$F_c^p(x_{ak}, x_{bk}, u_k, p, 0) = \frac{\partial f_c(x_{ak}, x_{bk}, u_k, p, 0)}{\partial p}. \tag{40}$$

Perturbation vectors δp_i are defined (analogous to $\delta \mu_i$):

$$\delta p_1 \equiv \begin{bmatrix} \Delta & 0 & 0 & 0 \end{bmatrix}' \tag{41}$$

$$\delta p_2 \equiv \begin{bmatrix} 0 & \Delta & 0 & 0 \end{bmatrix}' \tag{42}$$

$$\delta p_3 \equiv \begin{bmatrix} 0 & 0 & \Delta & 0 \end{bmatrix}' \tag{43}$$

$$\delta p_4 \equiv \begin{bmatrix} 0 & 0 & 0 & \Delta \end{bmatrix}', \tag{44}$$

by which the ith rows of partial derivatives of F_b^p and F_c^p are estimated:

$${}^iF_b^p \approx \frac{f_b(x_{ak}, x_{bk}, u_k, p + \delta p_i, 0) - f_b(x_{ak}, x_{bk}, u_k, p - \delta p_i, 0)}{2\Delta} \tag{45}$$

$${}^iF_c^p \approx \frac{f_c(x_{ak}, x_{bk}, u_k, p + \delta p_i, 0) - f_c(x_{ak}, x_{bk}, u_k, p - \delta p_i, 0)}{2\Delta}. \tag{46}$$

The quasi-optimal virtual mass estimator is summarized:

1. Initialize virtual mass estimate $\hat{p}$ (probably to zero) and 4×4 estimate covariance matrix P.

2. Propagate forward dynamics:

$$\begin{aligned} y_{ak+1} &= f_a(x_{ak}, x_{bk}, u_k, \hat{p}_k, 0) \\ y_{bk+1} &= f_b(x_{ak}, x_{bk}, u_k, \hat{p}_k, 0) \\ y_{ck+1} &= f_c(x_{ak}, x_{bk}, u_k, \hat{p}_k, 0). \end{aligned}$$

3. Propagate perturbed dynamics for y_b, y_c

$$\begin{aligned} y_{bk+1}^{i+} &= f_b(x_{ak}, x_{bk}, u_k, \hat{p}_k + \delta p_i, 0) \\ y_{ck+1}^{i-} &= f_b(x_{ak}, x_{bk}, u_k, \hat{p}_k - \delta p_i, 0) \\ y_{bk+1}^{i+} &= f_c(x_{ak}, x_{bk}, u_k, \hat{p}_k + \delta p_i, 0) \\ y_{ck+1}^{i-} &= f_c(x_{ak}, x_{bk}, u_k, \hat{p}_k - \delta p_i, 0). \end{aligned}$$

4. Estimate partial derivatives F_b^p and F_c^p. The ith rows are given:

$$\begin{aligned} {}^iF_b^p &\approx \frac{y_{bk+1}^{i+} - y_{bk+1}^{i-}}{2\Delta} \\ {}^iF_c^p &\approx \frac{y_{bk+1}^{i+} - y_{ck+1}^{i-}}{2\Delta} \end{aligned}$$

5. Calculate Gain:

$$\begin{aligned} \tilde{P}_{k+1} &= \hat{P}_k \\ K &= \tilde{P}_{k+1} \begin{bmatrix} F_b^p \\ F_c^p \end{bmatrix}' \left[W + \begin{bmatrix} F_b^p \\ F_c^p \end{bmatrix} \tilde{P}_{k+1} \begin{bmatrix} F_b^p \\ F_c^p \end{bmatrix}' + Q \right]^{-1} \end{aligned}$$

6. Update estimate $\hat{p}$:

$$\hat{p}_{k+1} = \hat{p}_k + K \left\{ \begin{bmatrix} y_{bk+1} \\ y_{ck+1} \end{bmatrix} - \begin{bmatrix} \hat{y}_{bk+1} \\ \hat{y}_{ck+1} \end{bmatrix} \right\} \tag{47}$$

7. Update covariance matrix P:

$$\hat{P}_{k+1} = \left\{ I - K \begin{bmatrix} F_b^p \\ F_c^p \end{bmatrix} \right\} \tilde{P}_{k+1}$$

The *control law* for the mass imbalance control system is simple when estimating the virtual masses. If the estimated position of the virtual masses is $\hat{p}$, the control law for moving the masses is:

$$p_{\text{commanded}} = -g\hat{p}, \tag{48}$$

where $0 < g \leq 1$.

In practice, IMU and accelerometer data are taken in batches and transmitted to the ground. The ground shall process the data and calculate a mass adjustment. Figure 4 demonstrates the system performance using the method of virtual masses.

THEORETICAL ISSUES OF OBSERVABILITY

Since many satellites have IMUs, their inclusion in the mass imbalance estimation system does not represent additional system hardware. Most satellites are not equipped with accelerometers so their inclusion does represent addition hardware with the usual concomitant increased cost, mass, power, and software. Elimination of the accelerometers is desirable so determination of their necessity is an important issue.

Simulation run using a single axis accelerometer demonstrated that only one axis was required. Simulation runs without accelerometers failed. To understand this failure a numerical observability test was performed.

Observability of discrete-time nonlinear systems is not addressed well in the literature with the the notable exceptions of one paper by Sontag [ref. 5] and one paper by Nijmeijer [ref. 6]. In [ref. 2], Grossman expands on the work of [ref. 6] and provides a simple test for observability for both forced and unforced systems.

Algebra for Discrete-time Nonlinear Systems

A causal discrete-time nonlinear SISO system is given by

$$x_{k+1} = f(x_k, u_k) \tag{49}$$

$$y_k = h(x_k), \tag{50}$$

where $x \in \mathcal{R}^n$.

The vector of n successive outputs is given by

$$\begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n-1} \end{bmatrix} = \begin{bmatrix} h(x_k) \\ h(x_{k+1}) \\ h(x_{k+2}) \\ \vdots \\ h(x_{k+n-1}) \end{bmatrix} = \begin{bmatrix} h(x_k) \\ h\,[f(x_k, u_k]))] \\ h\,[f(x_{k+1}, u_{k+1}))] \\ \vdots \\ h\,[f(x_{k+n-2}, u_{k+n-2})] \end{bmatrix}$$

$$= \begin{bmatrix} h(x_k) \\ h\,[f(x_k, u_k)] \\ h\,\{f[f(x_k, u_k), u_{k+1}]\} \\ \vdots \\ h\left[\mathcal{G}_f^{n-1}(x_k, u_k)\right] \end{bmatrix} \tag{51}$$

$$= \phi(x_k, u_k), \tag{52}$$

where the discrete-time analog of the Lie derivative is introduced and defined recursively

$$\mathcal{G}_f^0(x_k, u_k) = x_k \tag{53}$$

$$\mathcal{G}_f^n(x_k, u_k) = f\left\{\mathcal{G}_f^{n-1}(x_k, u_k), u_{k+n-1}\right\}. \tag{54}$$

(Note that the compact notation, $\mathcal{G}_f^n(x_k, u_k)$ hides the dependence of this quantity on $u_k, u_{k+1}, \cdots, u_{k+n-1}$)

Following [ref. 2], a system of dimension n is *strongly observable* over input sequence $\{u_k, u_{k+1}, \cdots, u_{k+n-1}\}$ iff

$$\operatorname{rank} \frac{\partial \phi(x_k, u_k)}{\partial x} = n. \tag{55}$$

Application of the Observability Test

Using this definition of strong observability, the observability of the virtual mass estimator was assessed numerically in the situation when only the IMU data is processed. Function f_c in (37c) is replaced by the rate forward propagation equation:

$$f_d(x_{ak}, x_{bk}, u_k, p_k, \mu) \equiv f_b(f_a(x_{ak}, x_{bk}, u_k, p_k, \mu), f_b(x_{ak}, x_{bk}, u_k, p, 0), u_k, p, 0). \tag{56}$$

The new synthesized measurement equation is given:

$$\begin{bmatrix} x_b \\ x_d \end{bmatrix} = \begin{bmatrix} F_b^p \\ F_d^p \end{bmatrix} \delta p \tag{57}$$

$$= \bar{F} \delta p \tag{58}$$

To numerically assess strong observability, the partial derivative matrix $\bar{F}$ of (58) is estimated (using the central difference technique applied previously) and the rank of $\bar{F}$ is determined from the condition number. In all simulation when accelerometers were absent, $\operatorname{rank} F < n$ indicating the system was not observable.

CONCLUSION

The Grossman/Friedland discrete-time parameter estimator has been successfully adapted to the mass imbalance estimation problem of a dual-spin satellite. When used to estimate the location of "virtual masses" and combined with a simple linear controller, the estimator provides a powerful technique to perform in-flight static and dynamic balance of a dual-spin satellite.

REFERENCES

[1] B. Friedland, *Advanced Control System Design.* Englewood Cliffs, NJ: Prentice-Hall, 1995.

[2] W. Grossman, *Observers For Discrete-Time Nonlinear Systems.* PhD thesis, New Jersey Institute of Technology, May 1999.

[3] W. Grossman, "Extension of the Friedland parameter estimator to discrete-time systems," *AIAA J. of Guidance, Control, and Dynamics*, vol. 20, no. 5, pp. 1047–1049, 1997.

[4] B. Friedland, "A nonlinear observer for estimating parameters in dynamic systems," *Automatica*, vol. 33, pp. 1525–1530, Aug. 1997.

[5] E. D. Sontag, "On the observability of polynomial systems, I:finite-time problems," *SIAM Journal Control and Optimization*, vol. 17, no. 1, pp. 139–151, 1979.

[6] H. Nijmeijer, "Observability of autonomous discrete time non-linear systems: A geometric approach," *International Journal of Control*, vol. 36, no. 5, pp. 867–874, 1982.

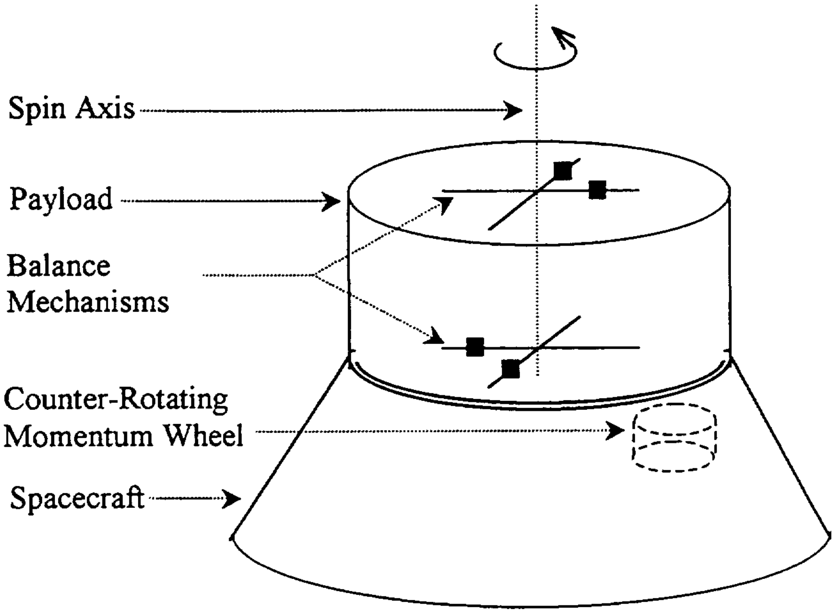


Figure 1: Dual spin satellite with mass imbalance control system.

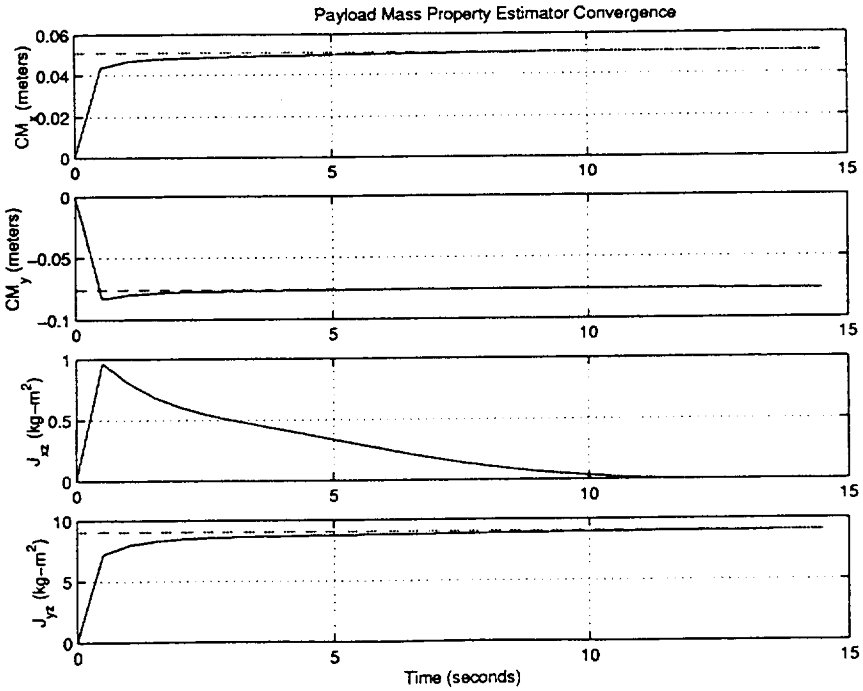


Figure 2: Convergence of the mass imbalance estimator.

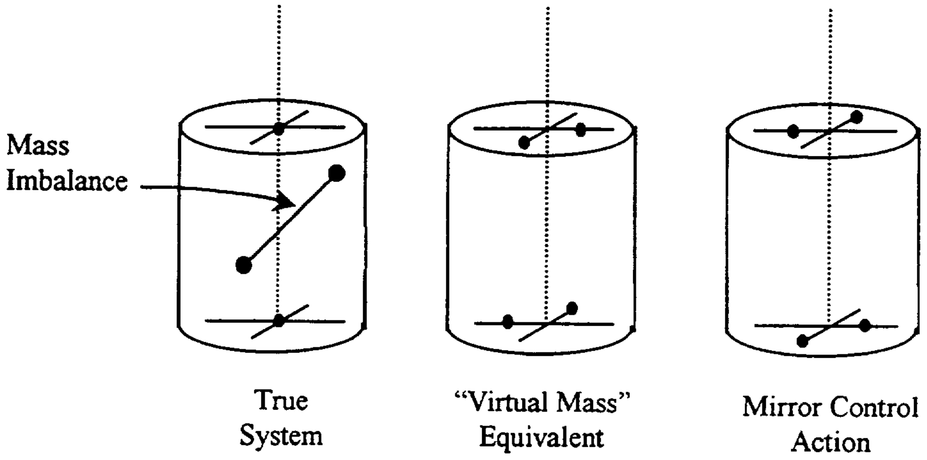


Figure 3: Mass imbalance control using method of "virtual masses."

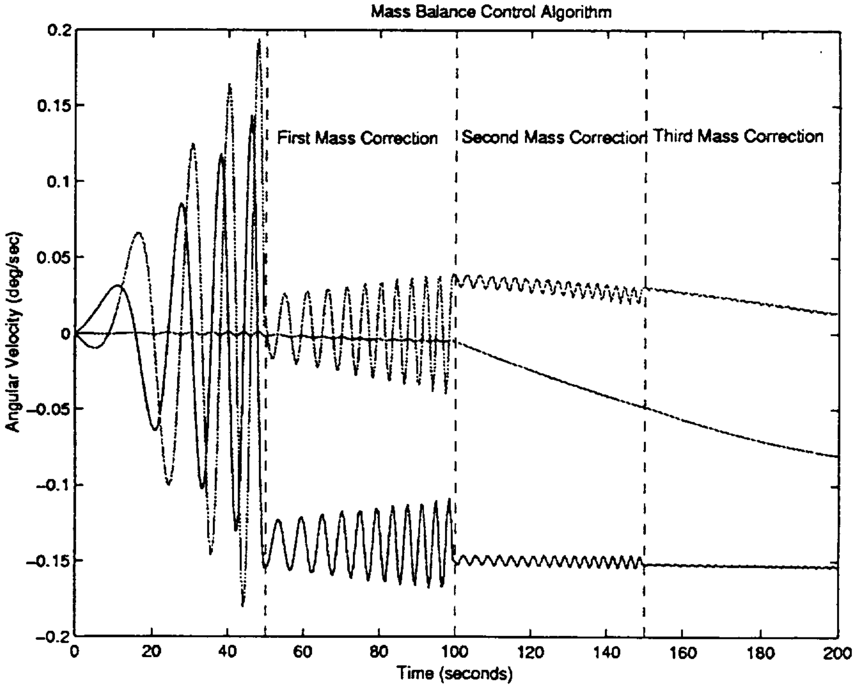


Figure 4: Reduction of attitude perturbations using "virtual mass" mass imbalance control system.

MANAGEMENT OF GUIDANCE, NAVIGATION AND CONTROL TECHNOLOGIES FOR SPACECRAFT FORMATIONS UNDER THE NASA CROSS-ENTERPRISE TECHNOLOGY DEVELOPMENT PROGRAM (CETDP)

Kathy Hartman, David Weidow
NASA Goddard Space Flight Center
Greenbelt, MD 20771

Fred Hadaegh
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109

ABSTRACT

Breakthrough technology development is critical to securing the future of our space industry. The National Aeronautics and Space Administration (NASA) Cross-Enterprise Technology Development Program (CETDP) is developing critical space technologies that enable innovative and less costly missions, and spawn new mission opportunities through revolutionary, long-term, high-risk, high-payoff technology advances. The CETDP is a NASA-wide activity managed by the Advanced Technology and Mission Studies Division (AT&MS) at Headquarters Office of Space Science. Program management for CETDP is distributed across the multiple NASA Centers and draws on expertise throughout the Agency.

The technology research activities are organized along Project-level divisions called thrust areas that are directly linked to the Agency's goals and objectives of the Enterprises: Earth Science, Space Science, Human Exploration and Development of Space; and the Office of the Chief Technologist's (OCT) strategic technology areas. Cross-Enterprise technology is defined as long-range strategic technologies that have broad potential to span the needs of more than one Enterprise. Technology needs are identified and prioritized by each of the primary customers. The thrust area manager (TAM) for each division is responsible for the ultimate success of technologies within their area, and can draw from industry, academia, other government agencies, other CETDP thrust areas, and other NASA Centers to accomplish the goals of the thrust area.

An overview of the CETDP and description of the future directions of the thrust area called Distributed Spacecraft are presented in this paper. Revolutionary technologies developed within this thrust area will enable the implementation of a spatially distributed network of individual vehicles, or assets, collaborating as a single collective unit, and exhibiting a common system-wide capability to accomplish a shared objective. With such a capability, new Earth and space science measurement concepts become a reality.

INTRODUCTION

As we look closer at the world we share, understanding the processes which define its composition and evolution, looking deeper into the universe to explore our neighbors and seeking to understand our origins, and expanding human presence within this universe, ever increasing demands are being placed on the technology we rely on to accomplish these tasks. Exploiting new vantage points, developing new sensing strategies, and implementing system-wide techniques which promote agility, adaptability, evolvability, scalability, and affordability are characteristic of the technological challenges faced and are representative of the significant leap beyond the current state of the art required. Efforts within the Distributed Spacecraft thrust area (TA) represent key technological developments which will support our ability to accomplish these tasks by succeeding traditional approaches to utilizing space and the limitations inherent to them, enhancing the ability of new and emerging technologies to further accomplish these challenges, and enabling entirely new measurement concepts to be realized.

Commonly referred to as formation flying, virtual platforms, virtual spacecraft, or virtual spacecraft bus architectures, these phrases refer to the introduction of a common distributed spacecraft control architecture within the framework established by multi-spacecraft missions. This architecture is characterized by interactions between spacecraft, cooperation between spacecraft, and common behavior across multiple spacecraft. Collectively, these attributes enable a distributed network of individual vehicles to act collaboratively as a single functional unit which exhibits a common system wide capability.

While rapidly approaching an era in which the capability to produce, deploy, and operate multi-spacecraft missions composed of highly capable, autonomous spacecraft is realized and readily available, these missions still remain for the most part characterized by collections of independent uncooperating assets. Unless facilitated through extensive ground support, this collection of vehicles (a.k.a. fleet) will continue to lack the capability required to interact among themselves, cooperate with each other as a single functional unit, and exhibit a common system wide behavior. To enable a distributed network of individual vehicles to act collaboratively as a single collective unit which exhibits a common system wide capability, revolutionary developments in how we conceptualize, design, implement, and operate missions are required and significant advances in the state of the technology are necessary. The study, development, and implementation of technologies to achieve a distributed spacecraft control architecture across multiple space assets forms the basis and focus for the Distributed Spacecraft thrust area.

OVERVIEW OF THE NASA CROSS ENTERPRISE TECHNOLOGY DEVELOPMENT PROGRAM

The NASA Cross-Enterprise Technology Development Program (CETDP) Plan[1] establishes operating principles and guidelines for thrust area management. The primary focus of the CETDP is to develop innovative technologies through basic research that address far-term scientific goals and spawn new measurement concepts and mission opportunities, and create new ways of doing Earth and space science for the Earth Sciences Enterprise (ESE), Human Exploration and Development of Space Enterprise (HEDS), and the Space Science Enterprise (SSE) communities. New mission opportunities are enabled through revolutionary long-term, high-risk, high pay-off technology advances. Investments are made in technology to support generic classes of challenging missions in advance of the development of specific missions. Co-funding from Enterprise sponsors is sought to advance the technology to the next maturity level for flight validation and use.

The CETDP is divided into two distinct elements: formulation and implementation. Each element has a single lead: a Formulator who plans the program while maintaining awareness of emerging technologies, and an Implementor who is responsible for program execution and performance monitoring. The program is coordinated across NASA at a high level by the NASA Technology Investment Board (CETIB) which consists of Directors from each of the participating Field Centers and primary customer Enterprise representatives. Program formulation is conducted annually but is flexible enough to incorporate new technology tasks at any time. The Formulator and Implementor work as a team to ensure overall CETDP success.

Technology research is conducted within thrust areas and managed by thrust area managers. These thrust areas will evolve over time to respond to future challenges. The current list of thrust areas includes: advanced power and onboard propulsion; breakthrough sensor and instrument component technology; distributed spacecraft; high rate data delivery; micro/nano sciencecraft; thinking space systems; surface systems; ultra-lightweight space structures and observatories; next generation infrastructure; and atmospheric and in-space systems. Technology challenges are extensive, diverse, and crosscutting in nature. Non-advocacy is used to ensure participation from the widest possible technology community. Peer reviews by government, industry, and academia help to establish and revise program goals, objectives, and the content of the focused technology areas. A series of open, technical workshops validate Enterprise requirements, identify technology challenges, and influence the comprehensive roadmap for each thrust area. Fair and open competition will be used to the fullest extent possible to develop revolutionary technologies.

Broadly advertised NASA Research Announcements (NRA) are the mechanism to solicit technology proposals on an annual basis. The NRA is developed with assistance from the TAMs and issued by the Director of AT&MS, who is the selection official. Non-advocate review panels whose membership is nominated by the TAMs, and approved by the Director rate the proposals. Multiyear grants or contracts to successful proposers will be issued and managed by the appropriate TAM.

FUTURE NASA MISSION ATTRIBUTES THAT DRIVE TECHNOLOGY CHALLENGES

Efforts within the Distributed Spacecraft thrust area (TA) represent key technological developments which will support our ability to accomplish these tasks by succeeding traditional approaches to utilizing space and the limitations inherent to them. Monolithic spacecraft, accommodating multiple experiments through a common spacecraft bus, are representative of traditional approaches to conducting Earth and space sciences (Figure 1). While generally successful in addressing the common needs across various instruments through shared resources, such "one solution fits all" approaches are now proving difficult to justify and even more difficult to implement in a climate of limited resources and heightened

[1] Cross-Enterprise Technology Development Program (CETDP) , "Program and Thrust Area Management", NASA Code SM, September 1998

expectations. The complexities inherent to large, multi-instrument spacecraft tend to reverberate throughout every aspect of the mission with each mission element, subsystem, and instrument impacting on every other and resulting in very high demands being placed on the spacecraft, ground systems, and operations. Long development times, sometimes exceeding a decade in length, and significant levels of funding, sometimes exceeding billions of dollars, become the norm. With the significant investments in time and money required, risk mitigation becomes a constant and overriding source of concern, adding to the time and money required to accomplish the mission. Emphasis switches from the science to be conducted to the spacecraft to be developed. In extreme situations, the complexities experienced and the time and money required are so severe that development efforts are never initiated or are terminated at some point during the program. Distributed spacecraft control technologies will enable a dramatic change in paradigm, replacing monolithic spacecraft hosting multiple instruments with a fleet of individual spacecraft each hosting individual instruments but controlled and behaving as a single functional unit.

As multi-spacecraft missions become commonplace, micro- and nano-satellite technologies mature, and the need for elaborate co-observing and multi-point measurement concepts increases, demands associated with the management and operation of these missions will become overwhelming. While capability is effectively distributed across a number of individual vehicles, potentially a large (>100) number of vehicles, the vehicles comprising the fleet remain independent assets which must be managed and operated as independent assets. Not only will this limit the ability to fully utilize the fleet to conduct the observing program for which it was developed, but also tremendous burden will be placed on the ground system and on operations personnel. As with traditional monolithic spacecraft approaches, this will result in significant investment in ground systems and operations being required, switching emphasis from the science to be conducted to the management of the multi-spacecraft fleet. Distributed spacecraft technologies will enable appropriate levels of interaction and cooperation between vehicles to autonomously manage the fleet as a single functional unit and obtain the desired behavior without major investments in ground systems and operations (Figure 2).

As new vantage points and sensing strategies are pursued to do what once were considered just a vision because the required technology is not on the immediate horizon, revolutionary approaches to how we conceptualize, design, implement, launch, and operate the next generation missions are required (Figure 3). The need to respond quickly, to react to changing conditions, to grow over time, to build off investments (starting small and growing), to spread costs, to reduce risk, and to demonstrate greater/faster return on investment are all characteristic of these systems. Enabling real-time decision making and control within fleets of spacecraft, distributed spacecraft technologies will promote significant increases in performance and complement a variety of other technology developments to support these missions.

RESEARCH SCOPE OF THE DISTRIBUTED SPACECRAFT THRUST AREA

A structured approach and framework is used to define the research scope of the Distributed Spacecraft thrust area. This approach begins by defining ***where we want to be*** in the future and establishing the expected state of the technology as derived from the advanced measurement concepts developed by the Earth and Space Science Enterprises. This will define the broad challenges and guiding principles upon which the thrust area will be based. Given this vision for the future, the current state of the art is assessed through a broad examination of technology development activities now underway or planned within NASA. This assessment defines ***where we are*** and establishes the technological foundation upon which the thrust area will be defined. Understanding where we are and where we want to be, the specific challenges associated and logical steps associated with ***how we get there*** are developed and the thrust area is defined. Finally, these are decomposed into specific technological challenges providing focus and direction to the program, characterizing the types of technology development activities required and the technology products to be developed, and forming the basis for proposal solicitation and selection activities associated with this thrust area.

Grand Vision - Expected State of the Distributed Spacecraft Technology

The scope of the Distributed Spacecraft thrust area is determined by many factors including the high-level mission challenges of the NASA Enterprises. Enabling technology concepts coupled with these mission challenges drive the research and technology development within this thrust area.

Current Enterprise measurement concepts are characterized by exciting and diverse science. And in turn, the technologies driving these new concepts are just as diverse and varied. Challenges include the need to respond quickly to new questions or new events. Global nowcasting, or predicting and monitoring Earth and space weather for disaster and environmental monitoring, will rely on correlated real-time measurements. Simultaneous multipoint probing at new and unique vantage points throughout the time-varying geospace is critical to achieving a global nowcasting capability (Figure 4). In addition, autonomous coordinated vehicles called planetary aerobots will move through atmospheres and study the environments of other worlds from higher vantage points, while at the same time exploring the surface and covering much more territory than planetary rovers (Figure 5). Multiple aerobots flying in formation offer significant improvement in 3-D stereo imaging of planetary bodies. Fleets of satellites around other planets can provide planetary services for mobile explorers, and deploy and support landers and samplers. Inflatable large-scale apertures on multiple spacecraft will be used to achieve desired interferometric observational baselines improving performance significantly. Space-based interferometry missions rely on the precision flying of a set of collector spacecraft with integrators to form the linear baseline (Figure 6). Achieving the needed precision alignment, maneuvering, and synchronized motion of a set of spacecraft remains a challenge.

Although the science is very different in these challenges, synergy exists across Enterprise concepts that employ the use of multi-platform, distributed assets. Distributed spacecraft control is a keystone crucial towards meeting these challenges. This technology will provide elements of the larger virtual infrastructure that allows adaptation to changing conditions and evolves over time to allow new ideas, new technology, and new capability, to simply join in the task at hand. Multiple vehicles will cooperate with one another, adapting to mission goals where vehicle pointing and positioning are managed collectively. Fleets evolve over time to extend and expand the capability of the system. Self-controlling, self-managing vehicles enable extensive co-observing programs to be conducted autonomously without complex multi-instrument observatories and extensive ground support. Significant improvements in space-based interferometry can be realized, increasing the number of instruments comprising the system and overcoming the challenges imposed when physical structures are used to establish, maintain, and control instrument separation.

Table 1 contains a list of current and planned satellite missions involving formation flying concepts. A designation is made to show whether the mission is Earth or space science driven. It is obvious from the number of missions included in this table that formation flying is a critical technology for future NASA missions.

Current State of the Technology

A necessary step in developing the technology direction is characterizing the state of the technology, or evaluating the status of the current research in the area of formation flying of satellites. The orbital mechanics and dynamics of multiple satellite systems in controlled formations has been studied for quite a while, but until recently the NASA culture was predominantly against the autonomous operation of satellites. The advent of shrinking budgets coupled with the push from the NASA Administrator to conduct innovative technology research that spawns new science measurement concepts have helped to open the door to the implementation of multiple satellite mission concepts. There has been an increase in proposed missions that utilize multiple satellites to accomplish mission goals, for both Earth and space science.

To begin to characterize the state of the technology, one must not neglect the fact that industry is developing constellations for mobile communications and data transfer. Motorola is the creator of the Iridium global, digital personal communications system with the goal of utilizing 66 satellites in six orbital planes. Motorola has plans for a follow-up system using 96 satellites. Another constellation under development is Globalstar, a consortium led by Loral Space and Communications. It has sixteen of the planned forty-eight satellites in orbit now. ORBCOMM has developed a system that uses low-Earth orbit satellites to provide monitoring, tracking, and messaging capabilities. Today, ORBCOMM has a constellation of 28 satellites providing near real-time communications availability. Teledesic is building a global, broadband "Internet-in-the-Sky" with a constellation of 288 low-Earth satellites. INMARSAT, an international consortium, is developing the ICO system, ten satellites at an altitude of 10355 km. Constellations using super-geosynchronous orbits providing regional coverage are under construction. Ellipso™ is a global, mobile personal communications by satellite (GMPCS) system with a unique, elliptical orbit configuration using two complementary sub-constellations totaling 17 satellites.

A distinction must be made between formation flying and constellation control. The Distributed Spacecraft thrust area is focusing on the control of multiple, cooperating satellites in autonomous formations that operate together to accomplish a variety of science objectives. Therefore, formation flying typically involves active, real-time, closed-loop control of these satellites in the formation. Formation flying can also be characterized as a combination of multiple assets, that is, space vehicles, sub-orbital balloons and surface robots, all operating autonomously together. Constellation control typically does not require this level of autonomous control. However, there are common subsystem elements like satellite cross-link communications and data transfer that are critical to both constellations and formations. In that sense the Distributed spacecraft thrust area will leverage off common subsystem elements that are commercially available, and not develop technologies that already exist.

Advances have been made in the control of multiple satellite formations. Some of the leading researchers in this field are: Goddard Space Flight Center (GSFC), Jet Propulsion Laboratory (JPL), Air Force Research Laboratory, Naval Research Laboratory, Johns Hopkins University Applied Physics Laboratory, Stanford University, Massachussetts Institute of Technology, University of California Los Angeles, and Space Products and Applications (SPA), Incorporated. The following includes highlights from ongoing formation flying research.

The NASA New Millennium Program (NMP) Space Technology (ST)-3 mission will demonstrate various elements of the technology required for space interferometry, including the Autonomous Formation Flying (AFF) sensor[1], and an autonomous reconfigurable formation control system[2,3].

ST-3 will consist of two spacecraft, each having a degree of autonomy, but both comprising a single instrument and constrained to move together in a relative distance of 50 to 1000 meters. In order to meet the mission's goals, the formation control system must control the distances between spacecraft within 1-2 centimeters, and the relative orientations of the spacecraft within 1 arcminute per axis. AFF borrows technology from the Global Positioning System (GPS), using measurements of both radio frequency (RF) carrier phase and a ranging code. Each spacecraft will have at least one transmitting antenna and two receiving antennas, operating at 30 Gigahertz with a code rate of 100 Mchips per second. Both spacecraft will be the collecting elements of the interferometer. They are roughly cubic in shape and have masses of about 150 kilograms. Each will use a mirror with a diameter of 12 centimeters to reflect the collected light (wavelengths 500 to 900 nanometers) to one of the spacecraft that will also serve as the combiner. Because of the modest collecting area, the faintest measurable sources will have visual magnitudes in the range 10 to 12. The interferometric baselines will vary in length from perhaps 50 to 1000 meters. During a planned lifetime of six months, the instrument will demonstrate its ability to point at specified targets, change baseline length, and maintain the formation at the required accuracy, as well as to find and track the interferometric fringes and report its measurements back to Earth. In the process, the instrument will measure the correlation amplitudes of 50 to 100 objects, mostly stars in our own galaxy. These amplitudes, in turn, will indicate the size and structure of the objects.

Clearly, monitoring and controlling the array configuration will be a crucial element in the operation of the ST-3 interferometer. That is, the collector spacecraft must be separated by the specified distance in the specified orientation and pointed toward the designated source in such a way that the two light paths reach the correct points at the combiner spacecraft. Furthermore, the total geometric lengths of the two light paths must be kept so nearly equal (within about 30 centimeters) that the optical path lengths can be equalized exactly by the optical delay line at the combiner. Incidentally, the need to maintain the array configuration without continual thruster firings mandates that the array operate away from strong gravity gradients—that is, beyond Earth orbit. As a result, ST-3 will operate in a solar orbit similar to the Earth's but trailing Earth approximately 0.1 astronomical unit (AU).

Conceptually, autonomous formation flying is a process in which an array of spacecraft makes continuous measurements of their "array configuration" and uses those measurements either to maintain an existing configuration or to move smoothly to a new one, all without external measurement or control. Generally speaking the array configuration would include both the distances between all pairs of spacecraft in the array and the orientations of the spacecraft in a coordinate frame defined by the array's internal geometry. From initialization to targeting, and then maneuvering to map the u-v plane for imaging in an interferometry mission for example, the formation control will experience significant changes to control requirements. Since one controller will not provide the formation stability and needed performance for various stages, a complete autonomous reconfiguarble control system will be implemented to switch between various models and various controllers. This new technology being developed currently at JPL and GSFC will serve as the basis for autonomous reconfiguarable control of future large formations and fleets for Earth and Deep Science missions.

The New Millennium Program Earth Orbiter (EO)-1 is scheduled for launch in December and will validate key aspects of NASA formation flying research. The enhanced formation flying (EFF) experiment on EO-1 includes autonomous maneuver decision making, planning and execution to maintain a one minute separation behind Landsat-7 to a tolerance of six seconds. Orbital altitudes for both spacecraft are 705 kilometers, polar, Sun-synchronous. This capability supports the development of multispectral paired-scene comparisons with the images from the enhanced thematic mapper instrument on Landsat-7. Formation flying architectures, strategies, and control approaches for stationkeeping , and target-chase scenarios will be validated. Four onboard formation flying algorithms developed by Goddard, JPL, SPA, Inc., and Microcosm, will be controlled under a flight executive named AutoCon during the mission[4]. The flight software is being developed by the Hammers Company and AI Solutions, Incorporated. The technology objectives are to validate aspects of autonomous control of multiple spacecraft; execute various autonomous orbit control algorithms; and build reusable, autonomous orbit control flight software.

With the successful demonstration of autonomous orbit maneuver planning and execution functions through the NMP EO-1 Formation Flying Experiment, a significant step towards autonomous distributed spacecraft control will have been taken. Building upon these capabilities, efforts are now focusing on methods to introduce varying levels of interaction, cooperation, and collective behavior within the control and management of spacecraft formations.

Interaction among spacecraft within a formation enables information related to the state of the formation to be shared and incorporated within the management of the formation. Introducing concepts such as decision making, hierarchical control, decentralized control, etc., within the management of the formation enables spacecraft within the formation to cooperate with one another. Such cooperative schemes will enable greater optimization of formation control parameters (e.g., fuel consumption, time to configure/re-configure formation) and increased flexibility and autonomy within the control of the formation (e.g., ability to adapt to changing conditions/requirements, fault detection/isolation/recovery). By combining appropriate levels of interaction and cooperation within the management of the formation, desired behaviors emerge and enable the formation to perform as a single functional unit (versus a collection of independent assets) in response to the objectives of the mission being flown.

Through the sponsorship of several NASA and non-NASA programs, the development of these capabilities are being pursued. Research and development efforts are being focused in several new areas. New sensing strategies and techniques are being developed to determine relative spacecraft positions and orientations within a formation. Inter-spacecraft communications strategies, techniques, and methods for distributing information within a formation are being studied and characterized. Formation control architectures, strategies, and control methods are being developed. Finally, the implementation of analysis and design tools and testbeds to support the development, validation, and demonstration of these capabilities are being supported. Several activities are representative of efforts now underway in these areas.

Building from experiences and lessons learned gained from the NMP EO-1 Formation Flying Experiment and recognizing the necessity for on-orbit demonstration of formation flying technologies, the Stanford University/GSFC Orion program was conceived as a low-cost, rapid means of studying the challenges posed by managing formations of spacecraft. The program was conceived as a means for supporting targeted technology development efforts through the implementation of end-to-end systems capable of supporting on-orbit engineering demonstrations of the technology[5]. Sponsored by the NASA Space Operations and Management Office (SOMO), Orion leverages off the extensive research already conducted in the areas of relative spacecraft navigation and formation control methods by Stanford University's Aerospace Robotics Laboratory (ARL) and the university class spacecraft development heritage of Stanford University's Space Systems Design Laboratory (SSDL). The Orion program is based on the concept of developing a university class spacecraft bus design capable of being implemented and replicated quickly in a university development environment (approximate 1-year development time), cheaply (approximately 250K per spacecraft), and possessing enough spacecraft capability and payload capacity (e.g., onboard computing, power, propulsion) to serve as a functional on-orbit testbed for the verification and demonstration of formation flying concepts and technologies. The Orion spacecraft are expected to be developed and flown in groups (2 or more) or in conjunction with other spacecraft. Design and prototyping of the first Orion vehicle will be completed in September of 1999 and the first Orion vehicle will fly in 2001 as part of the Department of Defense (DoD) University Nano-satellite Program. The first Orion spacecraft will study relative GPS navigation techniques and a number of formation control methods by flying in conjunction with the University Nano-satellites.

The NASA Earth Science Technology Office (ESTO) recently sponsored a broad suite of studies related to the application of formation flying technologies to Earth Science missions[2]. Included in these studies was a first attempt to fully understand and characterize inter-spacecraft communication needs for spacecraft flying in formation and assess design trades of RF-based communication approaches to address these needs. This study was conducted by the Johns Hopkins University Applied Physics Laboratory (JHU/APL). The Office of Space Science (OSS) Advanced Technology Development Program is now pursuing the design of a Crosslink Transceiver system with JHU/APL and ESTO is currently reviewing recommendations to support the implementation of this system. Also studied were deployment and initialization

[2] Earth Science Technology Office (ESTO), "ESTO Earth Science System Studies '98 for Flight Dynamics and Technology Requirements for Autonomous Formation and Constellation Flying", NASA Code Y, December 1998

strategies for spacecraft formations conducted by JHU/APL, general impacts of formation flying on science observations conducted by Virginia Tech University, and the integration of formation flying techniques and methods within existing mission analysis and design tools and testbeds conducted by AI Solutions Inc. and Stanford University.

Most notable of the non-NASA programs, the Air Force Research Laboratory (AFRL) has initiated a number of programs to develop formation flying technologies. The TechSat21 program is focusing efforts on the development of formation control algorithms[6]. The DoD University Nano-satellite Program, co-sponsored by the AFRL, Air Force Office of Scientific Research (AFOSR), and the Defense Advanced Research Projects Agency (DARPA), is sponsoring the development of nano-satellites (1 kg - 10 kg) by ten U.S. universities[7]. The University Nano-satellite Program has placed a special emphasis on the development and demonstration of formation flying technologies by the university teams. Recently, the NASA Cross-Enterprise Technology Development Program (CETDP) joined the DoD to further promote the development and demonstration of formation flying technologies under this program. Through the CETDP augmentation, funding is being provided to the universities to support teaming arrangements between universities, the establishment of university fleets (four individual fleets have been formed), and the development and accommodation of a variety of inter-spacecraft communications, relative spacecraft navigation, and formation control technologies.

In addition to these efforts, several other development activities are underway within NASA under the sponsorship of SOMO, CETDP, ESTO, OSS Explorers Technology Development Program, OSS Advanced Technology Development Program, Small Business Innovative Research (SBIR) Program, and the NASA Institute for Advanced Concepts (NIAC).

DISTRIBUTED SPACECRAFT THRUST AREA TECHNOLOGY INITIATIVES

A vigorous effort is needed to develop and demonstrate methodologies, algorithms, and the required component technologies to fully enable the control of distributed multiple assets for the accomplishment of diverse and varied science across multiple NASA Enterprises. This work includes systems studies to develop control architectures that are distributed across multiple satellites/assets, and specific technologies needed to accomplish the high-level thrust goals that will result in revolutionary advances over near-term and already planned technology programs. At the same time leveraging from other thrust areas, other NASA technology investments, and DoD, industry, etc. will ensure technology "gaps" are covered, duplication of effort is avoided, and partnerships are established which result in cost sharing and increased probability of success.

The Distributed Spacecraft thrust area supports the research and development of basic principles, concepts, approaches, techniques, spacecraft components, and strategies associated with the characterization and implementation of capabilities enabling interaction between vehicles, cooperation between vehicles, and the ability of a collection of vehicles to exhibit a common system wide capability (Figure 7). To accomplish this, a technology development program is envisioned which is composed of three general elements.

Central to the program are the capabilities, or themes, around which the technology development program will be organized: interaction between vehicles, cooperation among vehicles, and common system wide capability across vehicles. These themes represent the logical progression in capability necessary to advance the state of the technology, building off and extending current and expected technology development activities, to realize the goal of distributed spacecraft control (Figure 8). The second element of the technology development program focuses activities on broad, end-to-end systems engineering and technology demonstrations of the application of these capabilities in the development of measurement concepts for Earth and space sciences missions (Figure 9). A more thorough understanding of the needs associated with these capabilities will be formulated and will confirm or deny the type and level of capabilities required to support these applications. These efforts are expected to include a characterization of the type and extent to which common behavior need be exhibited across a fleet of spacecraft, developing an understanding of how spacecraft cooperate with one another and the level of cooperation necessary to support the common behavior required, and developing an understanding of why spacecraft need to interact with one another and the level of interaction necessary to support the level of cooperation required. The third element of the program targets the development of the specific technologies necessary to enable some element or combination of the capabilities required. The development of specific methods, techniques, strategies, components, and subsystems will be pursued.

After a preliminary assessment of the current and desired state of the technology, the high-level challenges associated with Distributed Spacecraft were determined to fall into two categories: technologies that are primary to the control of multiple assets, and supporting technologies.

- Primary to the control of multiple assets:
  Inter-spacecraft communication and synchronization of assets; fleet-level resource optimization (distributed network of assets across multiple platforms). Key to the achievement of this capability is the re-optimization of assets to target newly identified concerns or events and the ability to move or repoint the fleet in a timely manner; formation deployment, initialization, maintenance and control, and reconfiguration, on-orbit repair and upgrade; autonomous guidance, navigation and control across multiple assets (relative and absolute); fault protection methods; high performance sensors and actuators necessary to support autonomous formation control.

- Supporting technologies:
  Characteristic architecture/system attributes for distributed collaborating assets draw from many other thrust areas: onboard autonomy; integrated micro/nano avionics, sensors (subsystems on a chip), and detectors; learning, high-level reasoning/decision making; adaptation to new environments, adaptive reconfigurable systems; onboard high speed processing and pattern recognition with low-power and volume (e.g., compact optical processors); reduced reliance on ground for tracking, command and control and asset management; propulsion to support rapid mobility/reconfiguration and control of assets; distributed observation data collection, archival, dissemination and analysis

Distributed Spacecraft Products

The general challenges associated with formation flying are decomposed into specific technological challenges which characterize the type of technology development activities to be undertaken, that is, the technology products to be developed, and forms the basis for proposal solicitation and selection activities associated with this thrust area. A product breakdown structure represents the complete set of technology investments that fill the gap between where the current state of the technology is and where we want to be in the future. Figure 10 represents the preliminary Distributed Spacecraft product breakdown structure, listing high level investment areas. The following describes the overall context of each investment area.

In the sensors area the emphasis is on developing new sensing technologies for relative positions, velocities, and orientations among vehicles within the fleets, and perhaps even among cooperating fleets. The actuators area ensures distributed spacecraft control techniques are accommodated, that is actuators supporting orbit and attitude control systems must be precise, efficient, and support highly mobile vehicles. Telecommunications activities within this thrust area will concentrate on adapting communication technologies to new uses to support formation flying requirements. Immediate emphasis is placed on inter-spacecraft communications systems. Formation control is developing new control methods and architectures for all types of mission concepts. These control methods include fleet control paradigms and vehicle control algorithms. Onboard computing and data management ensures that distributed spacecraft control techniques are accommodated by providing high performance processors, high capacity data storage, and real-time distributed computing. And finally, tools and testbeds provide the required infrastructure for technology development, verification, and application. For example, mission analysis and design tools, flight software emulation environments, and component, subsystem, and system level verification are included.

These investment areas represent the initial set of product lines for the Distributed Spacecraft thrust area. The next step is to revisit in much more detail the specific mission needs for formation flying and map that back into the product breakdown structure, and a long-term technology roadmap. Customer reviews will be used to validate and revise the product breakdown structure. Overall success of the thrust area is determined by the progress of individual investment areas and the ability of the thrust area to meet customer needs while developing breakthrough technologies.

CONCLUSIONS

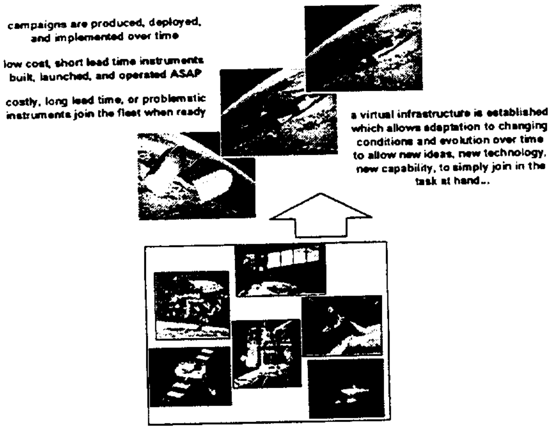
The primary focus of the CETDP is to fill the front end of the technology pipeline with technologies capable of supporting revolutionary advances. Formation flying concepts are appearing in many of the future NASA missions. And technologies that support formation flying are considered critical and vitally important to these varying mission concepts. The research within the Distributed Spacecraft thrust area is focused on the collaborative behavior of multiple space vehicles which form a distributed network of individual vehicles acting as a single functional unit while exhibiting a common system-wide capability to accomplish various mission goals. Technology investments within the Distributed Spacecraft thrust area will develop and validate methodologies, algorithms, and the required component technologies to fully enable the control of distributed multiple assets for the accomplishment of diverse and varied science across multiple NASA Enterprises.

REFERENCES

1. Lau, K., Lichten, S., Young, L., and Haines, B.,"An Innovative Deep Space Application of GPS Technology for Formation Flying Spacecraft," *Proc. of AIAA GN&C Conference*, pages 1-9, San Diego, August 1996.
2. Wang, P. K. C., and Hadaegh, F. Y.," Coordination and Control of Multiple Microspacecraft Moving in Formation", The Journal of Astronautical Sciences, 44(3): pages 315-355, September 1996.
3. Hadaegh, F. Y., Lu, W. M., and Wang, P. K. C., "Adaptive Control of Formation Flying Spacecraft for Interferometry", *Proc. International Federation of Automatic Control Conference*, Greece, July 1997.
4. Bauer, F., Bristow, J., Folta, D., Hartman, K., Quinn, D., and How, J., "Satellite Formation Flying Using and Innovative Autonomous Control System (AutoCon) Environment", *Proc. AIAA/AAS Astrodynamics Specialist Conference,* New Orleans, LA, August 11-13, 1997.
5. How, J., Twiggs, R., Weidow, D., Hartman, K., and Bauer, F.,"Orion: A Low-Cost Demonstration Formation Flying in Space Using GPS", *Proc. AIAA Guidance, Navigation, and Control Conference,* Boston, MA, August 10-12, 1998, AIAA paper 98-4398.
6. Das, A., Cobb, R., "TechSat21 - Space Missions Using Collaborating Constellations of Satellites", *Proc. AIAA/USU Conference On Small Satellites*, Logan, UT, August 31-September 3, 1998
7. Martin, M., Cobb, R., "University Nanosatellite Program", *Proc. IAF Synposium*, Redondo Beach, CA, April 19-21, 1999

Table 1. Satellite Missions Pertinent to Formation Flying and Distributed Spacecraft Concepts

Projected Launch Year	Mission Name	Mission Type
99	New Millennium Program (NMP) EO-1	Earth Science
01	University Nanosats/Air Force Research Laboratory	Technology Demonstrator
01	Gravity Recovery and Climate Recovery (GRACE)	Earth Science
02	Auroral Multiscale Mission (AMM)/APL (MIDEX)	Space Science
02	Lightweight Synthetic Aperture Radar (LightSAR)	Earth Science
03	New Millennium Program (NMP) ST-3	Space Science
03	New Millennium Program (NMP) ST-5	Space Science
03	Techsat-21/AFRL	Technology Demonstrator
04	Constellation-X	Space Science
05	Magnetospheric Multiscale	Space Science
05	Space Interferometry Mission (SIM)	Space Science
07	Global Precipitation Mission (EOS-9)	Earth Science
07	Global Electrodynamics	Space Science
08	Magnetospheric Constellation	Space Science
08	Laser Interferometric Space Antenna (LISA)	Space Science
09	DARWIN Space Infrared Interferometer/European Space Agency	Space Science
11	Terrestrial Planet Finder	Space Science
	Astronomical Low Frequency Array (ALFA)/Explorers	Space Science
	MAXIM X-ray Interferometry Mission	Space Science
05+	Leonardo (GSFC)	Earth Science
05+	Soil Moisture and Ocean Salinity Observing Mission (EX-4)	Earth Science
05+	Time-Dependent Gravity Field Mapping Mission (EX-5)	Earth Science
05+	Vegetation Recovery Mission (EX-6)	Earth Science
05+	Cold Land Processes Research Mission (EX-7)	Earth Science
05++	Submillimeter Probe of the Evolution of Cosmic Structure (SPECS)	Space Science
15+	NASA Goddard Space Flight Center Earth Sciences Vision	Earth Science
15+	NASA Institute of Advanced Concepts/Very Large Optics for the Study of Extrasolar Terrestrial Planets	Space Science
15+	NASA Institute of Advanced Concepts /Ultra-high Throughput X-Ray Astronomy Observatory with a New Mission Architecture	Space Science
15+	NASA Institute of Advanced Concepts /Structureless Extremely Large Yet Very Lightweight Swarm Array Space Telescope	Space Science



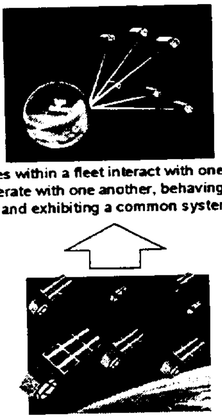
campaigns are produced, deployed, and implemented over time

low cost, short lead time instruments built, launched, and operated ASAP

costly, long lead time, or problematic instruments join the fleet when ready

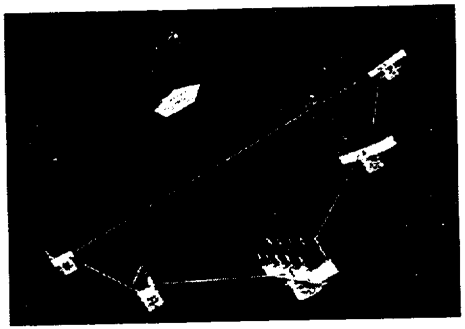
a virtual infrastructure is established which allows adaptation to changing conditions and evolution over time to allow new ideas, new technology, new capability, to simply join in the task at hand...

Figure 1. Succeeding Traditional Approaches



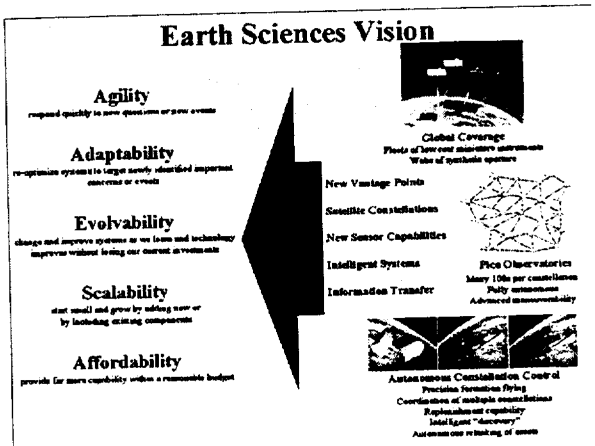
vehicles within a fleet interact with one another and cooperate with one another, behaving as a single functional unit and exhibiting a common system wide capability

Figure 2. Enhancing Current and Emerging Technologies



High precision virtual optical-truss systems providing the end-to-end planarity, centerline linearity, and optical path length control required to image the planetary systems of neighboring stars.

Figure 3. Enabling New Approaches



Earth Sciences Vision

Agility

respond quickly to new questions or new events

Adaptability

re-optimize systems to target newly identified important concerns or events

Evolvability

change and improve systems as we learn and technology improves without losing our current investments

Scalability

start small and grow by adding new or by including existing components

Affordability

provide for more capability within a reasonable budget

Global Coverage

Fleets of low cost miniature instruments

Webs of synthetic aperture

New Vantage Points

Satellite Constellations

New Sensor Capabilities

Intelligent Systems

Information Transfer

Pico Observatories

Many 10s per constellation

Fully autonomous

Advanced maneuverability

Autonomous Constellation Control

Precision formation flying

Coordination of multiple constellations

Replenishment capability

Intelligent "discovery"

Autonomous retasking of assets

Figure 4. Characteristics of the Earth Sciences Vision

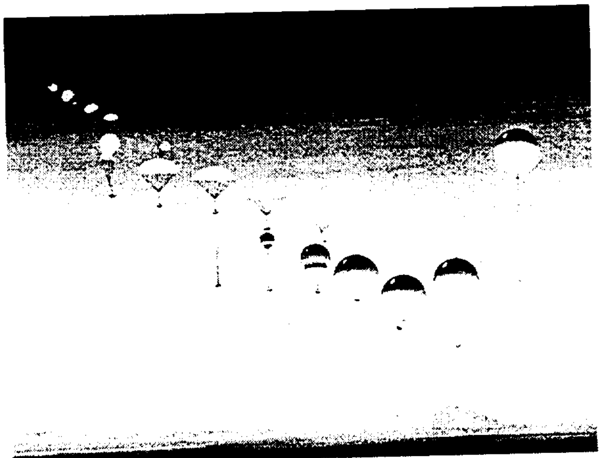


Figure 5. Formation Flying of Aerobots for Scientific Exploration

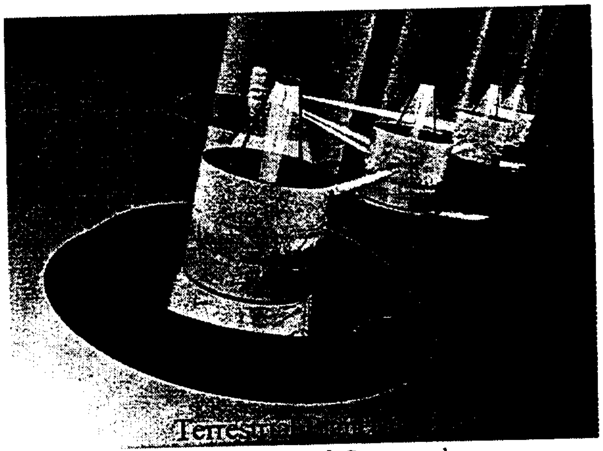


Figure 6. Five Spacecraft Separated Interferometer

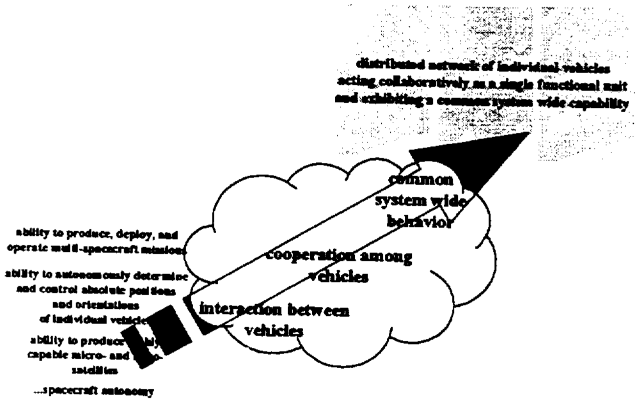

distributed network of individual vehicles
acting collaboratively as a single functional unit
and exhibiting a common system wide capability
common
system wide
behavior
cooperation among
vehicles
interaction between
vehicles
ability to produce, deploy, and
operate multi-spacecraft missions
ability to autonomously determine
and control absolute positions
and orientations
of individual vehicles
ability to produce highly
capable micro- and nano-
satellites
...spacecraft autonomy

Figure 7. Filling the Gap

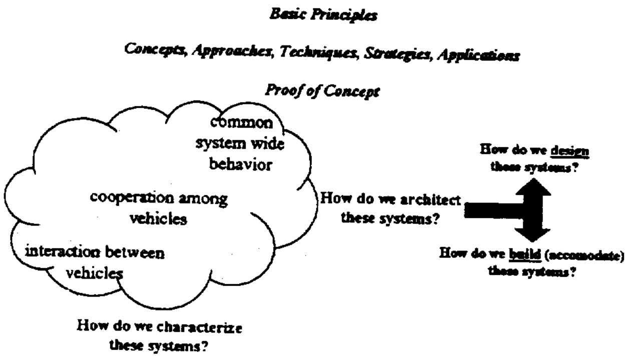


Figure 8. Thrust Area Goals

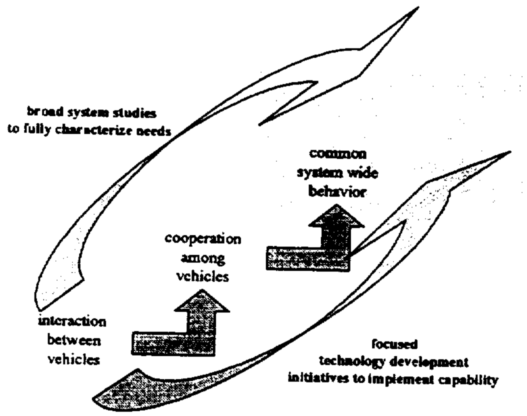


Figure 9. Thrust Area Focus and Approach

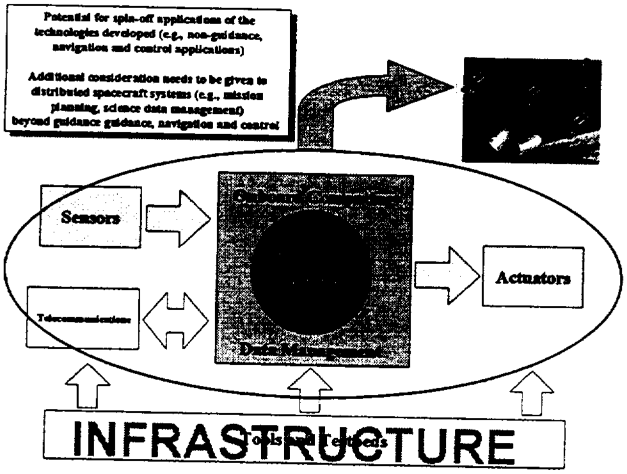

Potential for spin-off applications of the
technologies developed (e.g., non-guidance,
navigation and control applications)
Additional consideration needs to be given to
distributed spacecraft systems (e.g., mission
planning, science data management)
beyond guidance guidance, navigation and control
Sensors
Telecommunications
Data Management
Actuators
INFRASTRUCTURE
Tools and Testbeds

Figure 10. Distributed Spacecraft Technology Product Areas

DESIGN OF A FORMATION OF EARTH ORBITING SATELLITES: THE AURORAL LITES MISSION

Mark E. Hametz and Darrel J. Conway
AI Solutions, Inc.
10001 Derekwood Lane, Suite 215
Lanham, MD 20706

Karen Richon
National Aeronautics and Space Administration
Goddard Space Flight Center, Code 572
Greenbelt, MD 20771

ABSTRACT

The National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC) has proposed a set of spacecraft flying in close formation around the Earth in order to measure the behavior of the auroras. The mission, named Auroral Lites, consists of four spacecraft configured to start at the vertices of a tetrahedron, flying over three mission phases. During the first phase, the distance between any two spacecraft in the formation is targeted at 10 kilometers (km). The second mission phase is much tighter, requiring satellite interrange spacing targeted at 500 meters. During the final phase of the mission, the formation opens to a nominal 100-km interrange spacing. In this paper, we present the strategy employed to initialize and model such a close formation during each of these phases.

The analysis performed to date provides the design and characteristics of the reference orbit, the evolution of the formation during Phases I and II, and an estimate of the total mission delta-V budget. AI Solutions' mission design tool, ***FreeFlyer®***, was used to generate each of these analysis elements. The tool contains full force models, including both impulsive and finite duration maneuvers. Orbital maintenance can be fully modeled in the system using a flexible, natural scripting language built into the system. In addition, AI Solutions is in the process of adding formation extensions to the system facilitating mission analysis for formations like Auroral Lites. We will discuss how ***FreeFlyer®*** is used for these analyses.

THE AURORAL LITES MISSION

The mission analysis described in this paper was performed in support of the Auroral Lites Midex proposal submitted in August of 1998. The Auroral Lites concept encompassed multiple satellites flown in a rough tetrahedron formation in a near-polar highly eccentric orbit, thus combining the growing interest in constellation/formation flying with the intense scientific interest in the magnetospheric and electrical aspects of the auroral phenomena. The Auroral Lites goal was to provide the first unique space-time separated measurements in the auroral zones across multiple scales, exploring auroral plasmas across micro (kinetic), meso (regional) and global scales. This goal would further extend the scientific successes of past missions such as Dynamics Explorer, ISTP Polar, and FAST.

Auroral Lites is designed as a 1-year mission with a highly elliptical near-polar reference orbit that fully precesses in latitude and local time over a 1-year span. Measurements are taken from a series of inter-satellite separations ranging from 0.5 km to 100 km at apogee. Data collection nominally occurs from the middle magnetic latitudes to the magnetic poles. The data taken is uniformly distributed over the 4 auroral zone crossings for each orbit. The reference orbit is designed to maximize science return while meeting other mission constraints such as maximizing lighting and minimizing aerodynamic torques. This orbit accommodates launching four 195-kg satellites on a single Delta II 7320 launch vehicle.

Auroral Lites requires spinning satellites for optimal field alignment of the E-field wires, wire deployment and 3-D particle distribution. The spacecraft spins at 5-6 revolutions per minute, with a cartwheel like configuration ranging from a spin axis parallel to the orbit plane near the noon-midnight meridian plane to a spin axis normal to the orbit plane at the dusk-dawn plane. This configuration assures that the long electric field wires are nearly aligned to the Earth's magnetic field and that the solar cells are illuminated on the top deck of the spacecraft, with the spin axis nearly parallel to the Earth-Sun line.

The Auroral Lites mission uses a four-satellite constellation in order to generate uniquely separate space-time measurements in three dimensions. The identically equipped satellites perform simultaneous measurements of auroral, ionospheric and magnetospheric emissions. In addition, the satellites perform multiple stereo imaging of the local aurora in the spacecraft vicinity. The instrument complement includes instruments that perform AC and DC electric field and magnetic field measurements, electron and ion measurements and auroral imaging. A GPS receiver will be flown to provide sub-meter positioning and time synchronization between the satellites. The spacecraft use a hydrazine blowdown propulsion system that provides at least 95 m/s of delta-v for spacecraft spacing, orbit maintenance and spin control. The propulsion system consisted of eight 4.4 Newton thrusters and 11.6 kg of fuel per satellite. Six spin-control/delta-V thrusters are located on the cylindrical sides of the satellites, and two spin direction control thrusters are located on the aft deck. Figure 1 shows the spacecraft configuration.

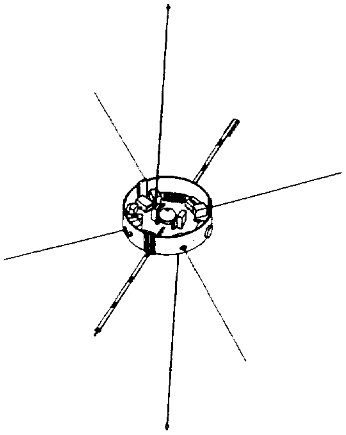


Figure 1: An Auroral Lites Spacecraft

In addition to the satellite measurements, intense ground campaigns of simultaneous magnetic field and radar imaging will be performed using observatories and magnetometers across the Northern Hemisphere, in Canada, Russia, Greenland and the United States. Orbital Sciences Corporation's (OSC's) Dulles, Virginia UHF ground station will provide command and telemetry support. An X-band downlink site in the continental US provides science data collection. OSC will perform mission operations, and Bowie State University will perform science data processing using the software and algorithms developed for the ISTP Cluster mission. The Flight Dynamics group at GSFC will perform the trajectory design and orbital maintenance at the direction of the Auroral Lites Science Team.

Auroral Lites is planned to launch directly into a 500 x 7000 km, 80-degree inclination orbit using a Delta II 7320. All four spacecraft are deployed from this launch vehicle. During the first 30 days of the mission, maneuvers will be performed to place the spacecraft in the initial constellation, a tetrahedron with approximately 10-km spacing. After 4 months of maintaining the 10-km spaced tetrahedron, the constellation will maneuver to a 0.5 km spaced tetrahedron, with a minimum inter-satellite separation of 0.25 km. This closest approach distance ensures that there is no possibility of satellite contact. After 4 months at the 0.5-km spacing, the satellites maneuver into their final constellation with 100 km nominal separation. An extended mission consisting of other constellation sizes may also be exercised if the spacecraft remain healthy and if sufficient fuel remains onboard each spacecraft.

We have modeled the Auroral Lites constellation using ***FreeFlyer®***, AI Solutions' flight dynamics tool. ***FreeFlyer®*** is a commercial off-the-shelf tool that supports flight dynamics across the mission lifecycle. The tool, described in the next section, allows for simultaneous modeling and control of multiple spacecraft, and provides unique visualization components to help the mission analyst design and monitor constellations of spacecraft.

FORMATION FLYING WITH *FREEFLYER®*

FreeFlyer® incorporates the key elements required to model and plan satellite operations for both individual spacecraft and multiple spacecraft formations and constellations. The system is robust, easy to set up and use, and flexible for use across the spectrum of Earth-orbiting satellites. It includes precision modeling of the satellite's orbit, including effects from the Earth's oblate gravitational field, gravitational effects from the Sun, Moon, and solar system planets, solar radiation pressure, and atmospheric drag. The system models both finite and impulsive maneuvers and computes the effects of these maneuvers on both the satellite's orbit and its mass properties. It is an object-oriented software system that is designed to support extension to new problem regimes.

FreeFlyer® provides the tools required to model and maintain both close formations of spacecraft and constellations of spacecraft supplying whole Earth coverage. The user can use the software to initialize the satellite configuration and model the evolution of this configuration across the lifetime of the mission. The system propagates each satellite independently, maintaining the time synchronization between the satellites during the orbital evolution. Orbital maintenance can be fully modeled in the system using a flexible, natural scripting language. ***FreeFlyer®*** contains interfaces to external processes, so that calculations not modeled in the core system can be modeled externally and incorporated into the running system.

Formation Visualization

FreeFlyer® provides three different trajectory views of a formation's orbit. The user can view the formation's groundtrack, its three-dimensional trajectory around the Earth, or the motion of its trajectory relative to a reference orbit. The latter of these is the most germane for the Auroral Lites mission. The software includes a visualization component designed to display relative motion called the control box. The control box provides a mechanism for displaying the along-track, cross-track, and radial offsets in position of one or more spacecraft relative to a nominal position. The control box object behaves like a spacecraft: it is propagated through time, and is used to model the ideal orbit for the mission. The user displays the control box as a rectangular cube with the spacecraft's relative position modeled inside. The control box view provides an easily interpreted view of the relative motion of multiple satellites with respect to one another.

Figure 2 shows the control box for the Auroral Lites mission. For this phase of the mission, the spacecraft are required to maintain a formation in which no spacecraft approaches closer than 5 km from any other, and never further than 30 km from any other, meeting the nominal 10-kilometer spacing requirement. The control box view allows users to set parameters that simplify visualization of such constraints. Figure 2 is set to display the reference trajectory as a cube with side lengths set to 30 km on each edge. The satellites are displayed in this enhanced version of the control box as spheres with radius 2.5 km. Thus if two spheres intersect, the satellites are too close together, and if any two satellites are separated by a distance greater than the length of one of the cube sides, the satellites are too far apart. The control box display is updated at each time step of the orbital evolution, so the analyst gets immediate visual feedback regarding the status of the formation constraints.

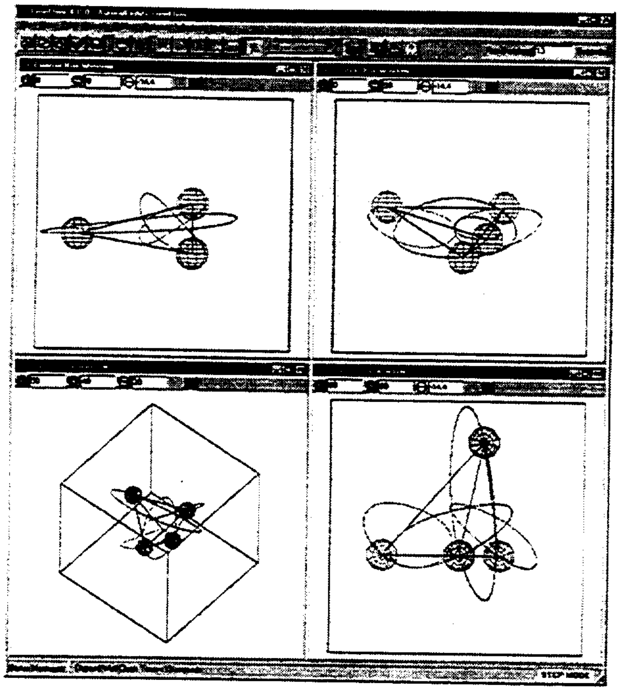


Figure 2: Visualization of Auroral Lites in the ***FreeFlyer®*** Control Box

Formation Monitoring

We used ***FreeFlyer®***'s internal calculations of orbital parameters to analyze the behavior of the Auroral Lites formation over time. The parameters calculated by ***FreeFlyer®*** cover most mission requirements; for Auroral Lites the shadow, the contact and formation separation parameters were used extensively to design the formation.

In the next section we discuss how the core parameter calculations provided by the software were used to set the initial orbital elements for each satellite in the tetrahedron, and to monitor the inter-satellite spacing for the formation.

MISSION ANALYSIS RESULTS

This section summarizes the results of the Auroral Lites analysis that have been performed in support of the MIDEX mission proposal. The results include the following:

- *Reference orbit design and characteristics*
- *Formation flying for Phases I and II*
- *Estimated delta-V budget*

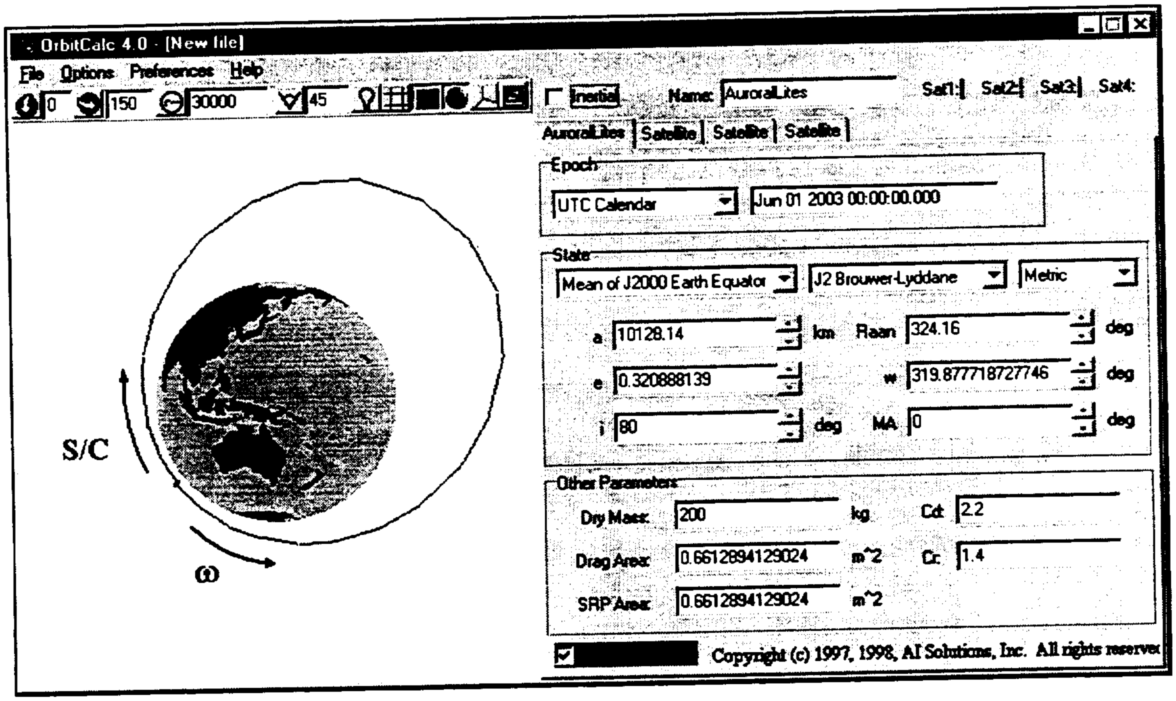
Reference Orbit Design and Characteristics

The reference orbit is designed as a 500 by 7000-km orbit at an inclination of 80 degrees. The target launch date is June 1, 2003. The first 30 days of the mission will be used for the formation deployment and systems checkout. The science portion of the mission will then commence for 1 year for a total mission duration of 13 months. The natural evolution of the orbit will cause the argument of perigee to rotate opposite to the direction of motion at the rate of approximately 1-degree per day. The Mean Local Time (MLT) of the ascending node will rotate approximately 34 hours in 1 year.

With these design conditions, the Project Team selected the initial orbit orientation as described below. The argument of perigee was chosen to position apogee at 40 degrees north latitude, on the daylight side of the orbit. The initial node was chosen to set the MLT of the ascending node to 5 a.m. With this configuration, the orbit plane will evolve towards a noon-midnight orientation (towards the Earth-Sun line) while apogee rotates towards the North Pole. When the science phase of the mission begins, apogee will still be on the daylight side allowing northern daylight targets to be captured at high altitudes early in the mission. As the mission progresses, apogee will rotate to the night side where northern night targets will be observed.

Initially, the southern latitudes will be studied at low altitudes. As the orbit evolves, higher altitude science will be obtained for the middle portion of the mission. Towards the end of the mission, the orbit will rotate near the initial orientation for another northern campaign.

The initial state used in the following analysis of the reference orbit is shown in Figure 3, along with the physical spacecraft parameters. All orbit propagation and data generation is performed using ***FreeFlyer®*** with an 8x8 degree and order Joint Gravity Model-2 (JGM2) geopotential model, a Jacchia-Roberts atmospheric drag model using nominal solar flux activity predictions, Sun and Moon gravitational effects, and solar radiation pressure.



0 | 150 | 30000 | 45

□ Inertial

Name: AuroralLites

Sat1: | Sat2: | Sat3: | Sat4:

AuroralLites | Satellite | Satellite | Satellite

Epoch

UTC Calendar | Jun 01 2003 00:00:00.000

State

Mean of J2000 Earth Equator | J2 Brouwer-Lyddane | Metric

a	10128.14	km	Raan	324.16	deg
e	0.320888139		w	319.877718727746	deg
i	80	deg	MA	0	deg

Other Parameters

Dry Mass:	200	kg	Cd:	2.2
Drag Area:	0.6612894129024	m^2	Cr:	1.4
SRP Area:	0.6612894129024	m^2		

☑ Copyright (c) 1997, 1998, AI Solutions, Inc. All rights reserve

Figure 3: Reference Orbit Initial State

Once the orbit was chosen, ***FreeFlyer®*** was used to produce various ephemerides, plots, and data files for use by spacecraft subsystem designers. The detailed results of these analyses are shown in the pages below.

Mean Local Time

The power subsystem engineers used MLT/beta angle evolution and eclipse information to determine appropriate battery and array sizing. The evolution of the mean local time (MLT) of the ascending node (Figure 4) defines the orbit plane orientation with respect to the mean Sun position. A noon/midnight MLT places the mean Sun vector in the orbit plane, while a 6 a.m./6 p.m. MLT places the orbit-plane perpendicular to the mean Sun vector. As Figure 4 shows, the MLT is near midnight at the outset of the mission to provide optimum lighting conditions for the northern latitude targets observed at high altitudes. Conversely, the rotation of the line of apsides is coordinated with the MLT evolution such that southern latitudes will be observed at high altitudes at near-midnight lighting.

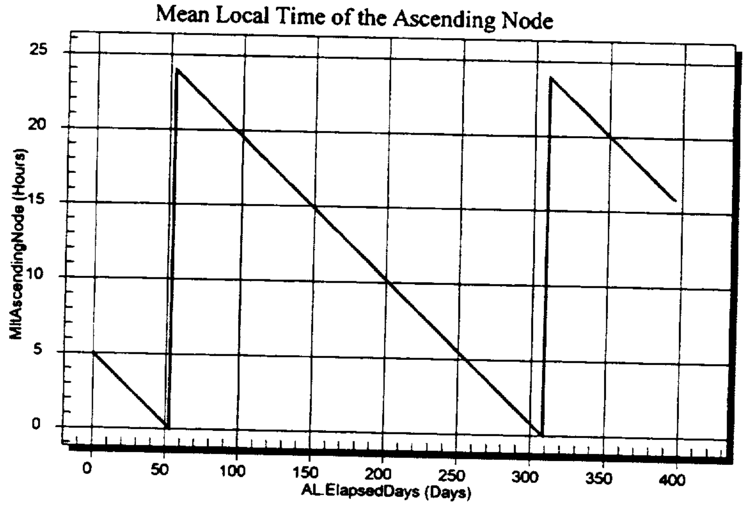


Figure 4: Evolution of the MLT of the Ascending Node

Shadow Duration

Figure 5 shows the shadow occurrences for the reference orbit for Earth eclipses. The shadow season lasts approximately 100 days. The maximum shadow duration encountered was 35 minutes. The shadow seasons coincide with the MLT evolution shown above. The orbit falls into shadow when the orbit MLT falls away from a 6 am/6 pm orientation. The longer shadow periods will occur when apogee has rotated to the night side of the orbit.

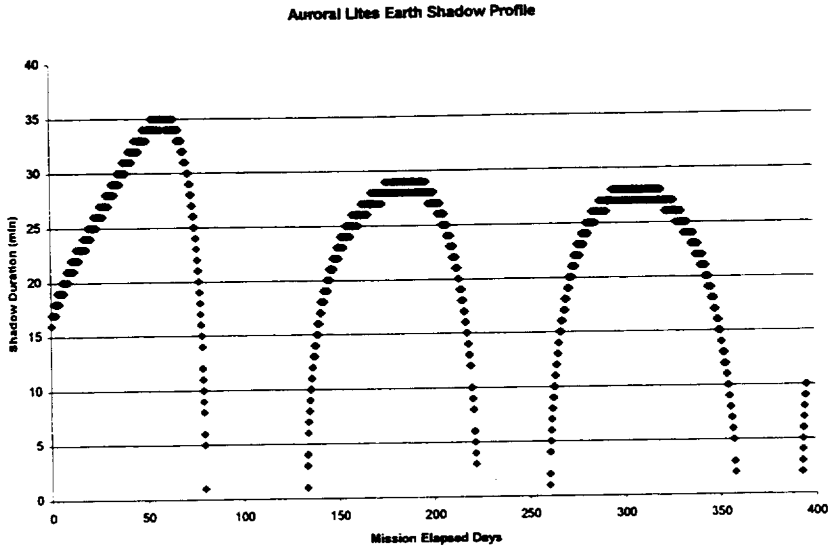


Figure 5: Shadow Durations for Auroral Lites

Orbit Stability

For eccentric orbits, third body effects can cause large fluctuations in perigee height based on the phasing of the satellite orbit relative to the Moon. The Auroral Lites orbit was tested for this sensitivity by modeling the evolution of the orbit for launch dates over a 30 day period at 5 day increments (June 1 to June 30, 2003). Both the apogee and perigee heights are stable for the full range of dates tested. Perigee height is presented as a function of time in Figure 6.

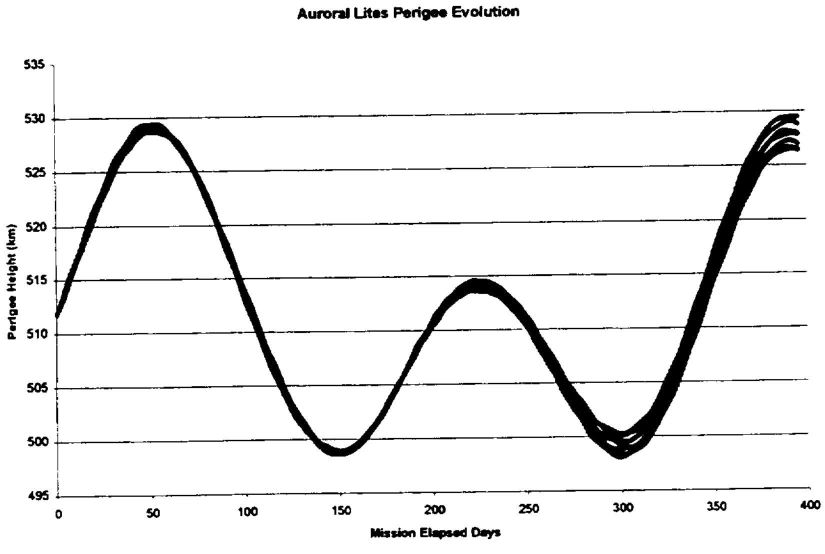


Figure 6: Evolution of the Perigee Height

Wallops Island Contacts

Station contact information was provided to determine the communications subsystem and the onboard data storage unit characteristics. Figures 7 and 8 show the contact history for this orbit. At the start and end of the mission, there are a large number of quality passes with large pass duration. During the middle of the mission, the number and duration of the passes diminishes significantly. This widely changing pattern is again due to the orbital evolution. The Wallops passes are at a minimum when apogee has reached its southern-most point in the evolution. Therefore, to maintain high numbers of long duration passes, a southern latitude ground station should be utilized.

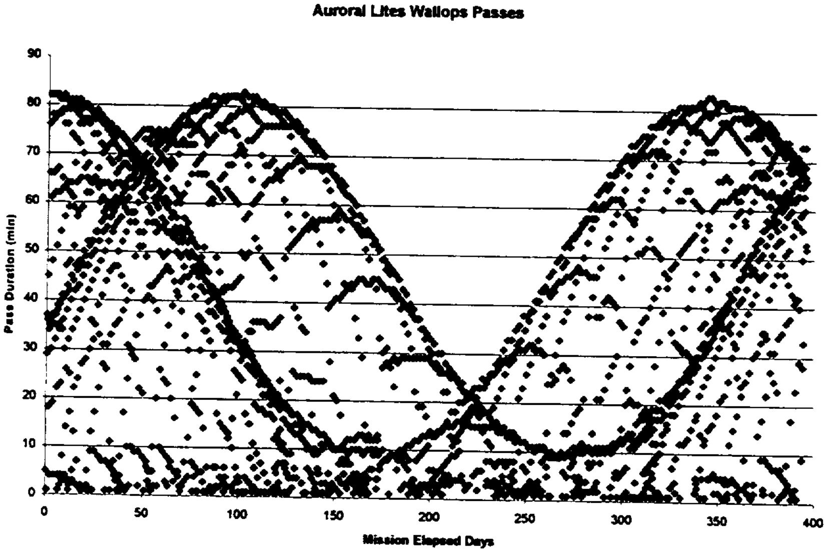


Figure 7: Pass Duration history at Wallops Island

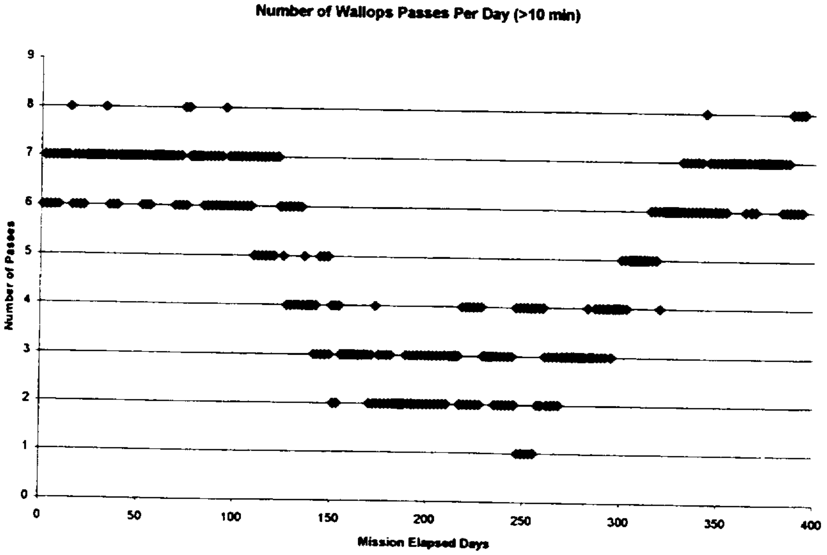
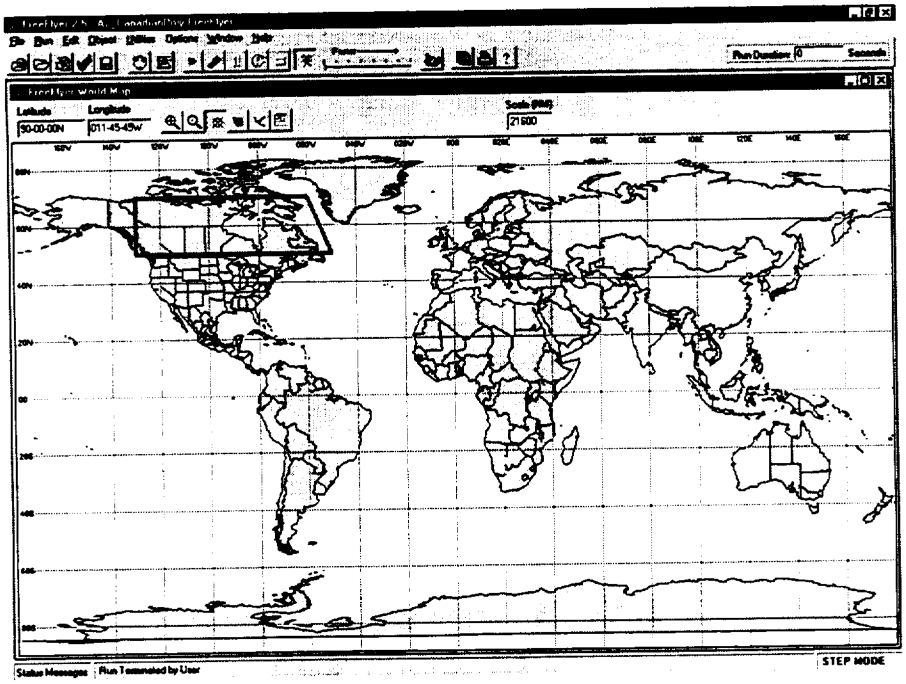


Figure 8: Pass Count at Wallops Island

Ground Correlation

FreeFlyer® was also used to assist the science team in predicting the science data characteristics during the mission. One example utilized the capability to predict when a spacecraft's subsatellite point is within a user-defined polygon region. Early in the mission, night coverage will be obtained from the Auroral Lites satellites at high latitudes. Concurrently, ground observations can be correlated with the on-orbit science. Figures 9 and 10 show the altitudes that will be sampled, along with the mean local time of the observations for nighttime passes over the Canadian region outlined in the polygon in the 2D map below.



Run Duration 0 Seconds

Latitude 50-00-00N

Longitude 011-45-45W

Scale (NM) 21600

Status Messages | Run Terminated by User | STEP MODE

Figure 9: Canadian Contact Region

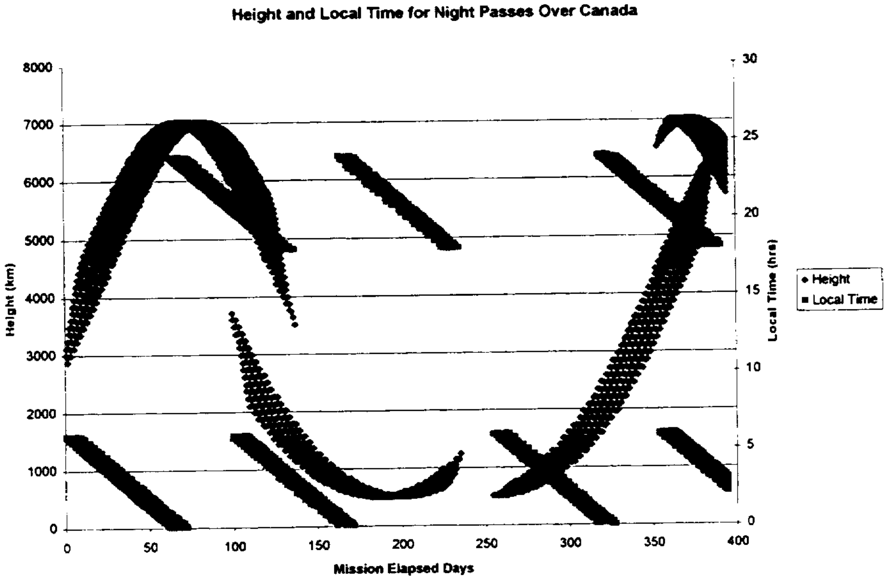


Figure 10: Data Passes over Canadian Contact Region

Formation Flying

The four identical spacecraft of the Auroral Lites constellation fly in a tetrahedron formation. During the first 30 days, in the checkout phase, the spacecraft will be separated from one another into the formation at a nominal spacing of approximately 10 km. This first spacing, Phase I, will be flown for 4 months. Then, in Phase II, the separation will be reduced to 0.5 km for another 4 months. For Phase III of the mission, the separation between the spacecraft will increase to a nominal 100-km spacing for another 4 months.

The first two mission phases were studied to derive orbit states that satisfied the mission constraints. Since the tetrahedron formation cannot be constantly maintained without constant thrusting, a formation that reasonably kept the multi-dimensional shape was designed. The interrange distances over the entire orbit were kept to within 3 times the target range value (Phase I: 10 - 30 km, Phase II: 0.5 – 1.5 km, Phase III: 75-300 km).

The basic design of the formation is as follows. Three of the spacecraft were kept in the reference orbit plane. Spacecraft 1 and 2 were placed slightly below the reference orbit, one leading the other. Spacecraft 3 was placed slightly higher and between 1 and 2 to complete the triangle. Spacecraft 4 was placed out of the orbit plane to complete the pyramid.

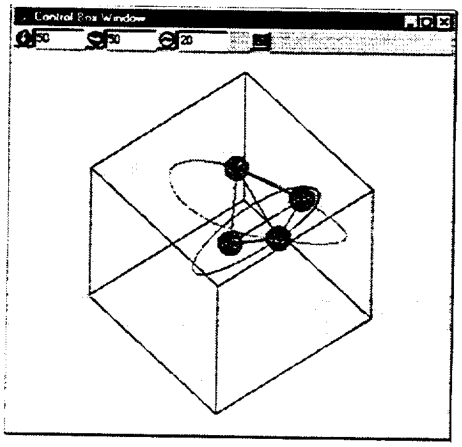


Figure 11: Relative satellite motion for Auroral Lites

Figure 11 shows the relative motion of the spacecraft over one orbit using ***FreeFlyer®***'s Control Box. The base spacecraft is fixed at the center of the box, while the other spacecraft are shown moving relative to it. Spacecraft 2, located behind the central spacecraft, moves predominantly in the along track direction; its orbit trace appears as a straight line. The third spacecraft starts above the first two, and loops below them so as to maintain orbital phasing. The fourth spacecraft differs in plane through a nodal difference and therefore moves side-to-side over each orbit.

Instantaneous views of the relative locations of the spacecraft as they travel clockwise around the orbit are shown in Figure 12. The individual control box images are taken at true anomalies of 0, 90, 180, and 270 degrees.

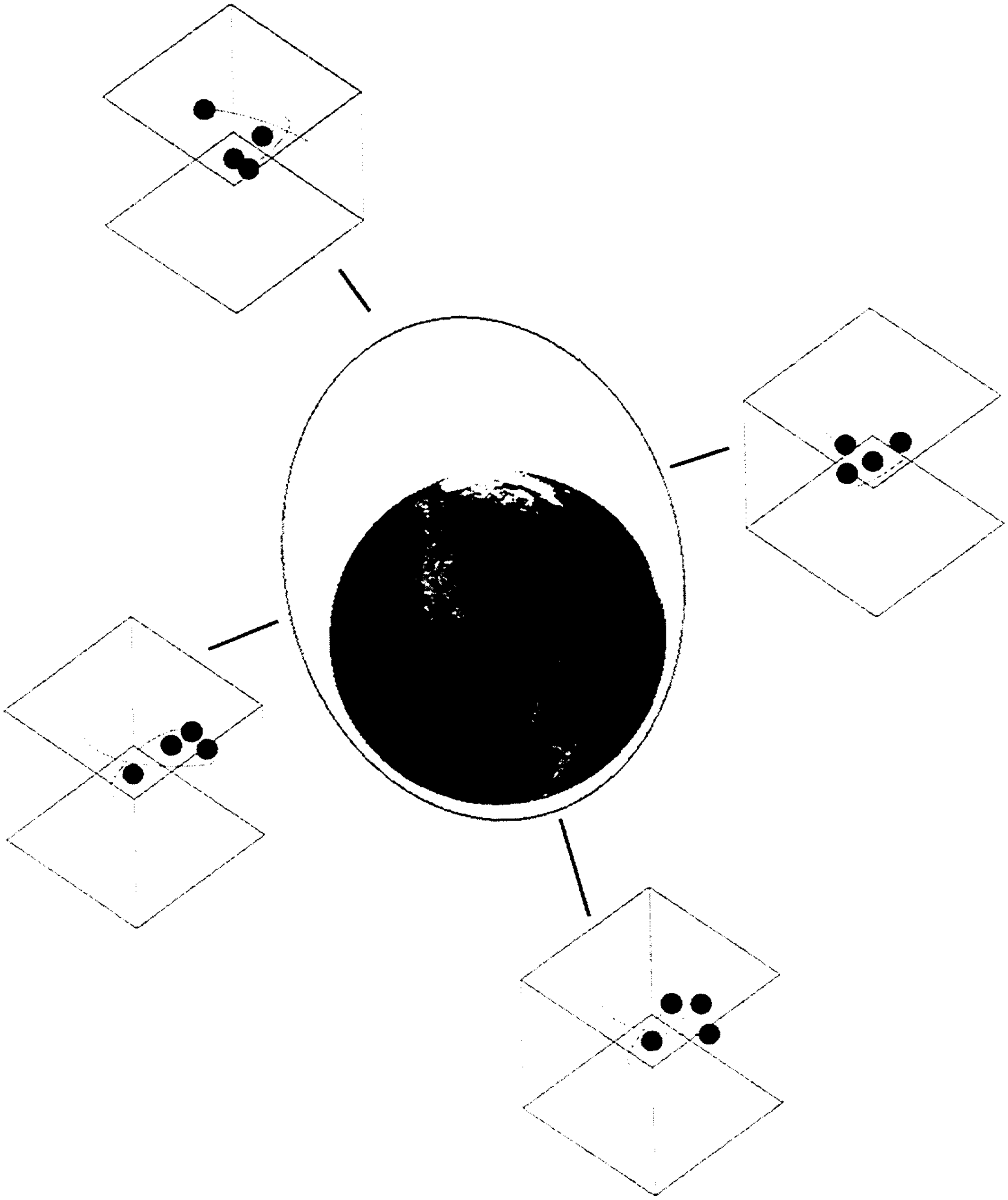


Figure 12: Satellite Positions around the Auroral Lites orbit

The orbital states tabulated below provide consistent interrange values of either between 5-27 km (Phase I), 0.25-1.5 km (Phase II), or 75-225 km (Phase III) for 30 days without formation maintenance maneuvers. All of these states are listed as Mean of J2000 Earth Centered Inertial Keplerian elements. The epoch for each state is June 1, 2003 00:00:00.000 UTC.

Spacecraft 1:

Parameter	Phase I (10 km)	Phase II (0.5 km)	Phase III (100 km)
A	10123.145	10123.142	10123.145
E	0.321046473	0.321046473	0.321046473
I	80.0	80.0	80.0
RAAN	309.160505	309.160505	309.160505
W	319.92063256	319.92063256	319.92063256
TA	0.0303	0.0303	0.0303

Spacecraft 2:

Parameter	Phase I (10 km)	Phase II (0.5 km)	Phase III (100 km)
A	10123.0899	10123.13916	10122.591
E	0.321046473	0.321046473	0.321046473
I	80.0	80.0	80.0
RAAN	309.160505	309.160505	309.160505
W	319.84063256	319.91653256	319.12063256
TA	0.0303	0.0303	0.0303

Spacecraft 3:

Parameter	Phase I (10 km)	Phase II (0.5 km)	Phase III (100 km)
A	10123.129	10123.14097	10122.98
E	0.320256205	0.32099708	0.314131626
I	80.0	80.0	80.0
RAAN	309.160505	309.160505	309.160605
W	319.88063256	319.9180825	319.52063256
TA	0.0303	0.0303	0.0303

Spacecraft 4:

Parameter	Phase I (10 km)	Phase II (0.5 km)	Phase III (100 km)
A	10123.1239	10123.14055	10122.92
E	0.320552556	0.321021777	0.31758905
I	80.0	80.0	80.0
RAAN	309.0400505	309.1549766	308.560605
W	319.88063256	319.9180825	319.52063256
TA	0.0303	0.0303	0.0303

Table 1: Orbital States for Auroral Lites Spacecraft

Delta-V Budget Estimation

Based on the orbital elements derived in the previous section, a delta-V budget was estimated for the mission and is shown below. Assuming the spacecraft began in identical orbits (the reference orbit), the delta-V was calculated to move between the mission phases to the target states to begin each phase. Although no formation maintenance strategy was studied, a stationkeeping budget was estimated based on a reasonable amount of in- and out-of-plane maneuvering that would be expected along with two "rules of thumb". First, the farther the spacecraft are set apart from each other, the more unequal the forces that affect each spacecraft, causing more frequent maneuvering. Second, since the spacecraft are maneuvering differing amounts, the difference in ballistic coefficients is constantly growing. This implies that the amount of maneuvering required will increase over the mission lifetime.

The largest delta-V required is for Spacecraft 4. This is primarily due to the out-of-plane maneuvering that is required. The smallest delta-V required is for Spacecraft 1, which has minimal in-plane maneuvering to perform in the initial formation configuration, and is then held as the reference spacecraft for the final phases. If the spacecraft need to be actively reentered, the approximate delta-V cost is an additional 100 m/s per spacecraft.

	Spacecraft 1 Delta-V (m/s)	*Spacecraft 2 Delta-V (m/s)*	*Spacecraft 3 Delta-V (m/s)*	*Spacecraft 4 Delta-V (m/s)*
PHASE I				
Configure to 10 km				
In-plane adjustments	1.64	3.30	2.92	1.94
Out-of-plane adjustments	0.0	0.0	0.0	10.0
Maintain 10 km Formation	2.0	2.0	2.0	2.0
Subtotal	***3.64***	***5.30***	***4.92***	***13.94***
PHASE II				
Configure to 0.5 km				
In-plane adjustments	0.0	1.17	3.23	2.33
Out-of-plane adjustments	0.0	0.0	0.0	9.47
Maintain 0.5 km Formation	2.0	2.0	2.0	2.0
Subtotal	***2.0***	***3.17***	***5.23***	***13.8***
PHASE III				
Configure to 100 km				
In-plane adjustments	0.0	16.62	30.99	19.63
Out-of-plane adjustments	0.0	0.0	0.0	49.8
Maintain 100 km Formation	10.0	10.0	10.0	10.0
Subtotal	***10.0***	***26.62***	***40.99***	***79.43***
TOTAL	**15.64**	**35.09**	**51.14**	**107.17**

Table 2: Estimated Spacecraft Delta-Velocity Budget

CONCLUSIONS

Auroral Lites is a proposed mission that effectively demonstrates the need for formations of spacecraft in order to advance our understanding of the auroral regions. The mission analysis to date includes precision modeling of the formation of spacecraft that collect the data, and analysis of the formation evolution over time. We have performed these analyses using the modern, interactive tool ***FreeFlyer®***. The resulting analysis demonstrates the constraints and evolution of a tight formation of spacecraft tailored to observing the Earth's auroral regions.

References

1. S. A. Curtis, "Auroral Lites, A MIDEX International Multiprobe Mission Proposal," August 1998.

2. M. E. Hametz, "Auroral Lights Analysis", AI Solutions, Inc. memorandum, August 1998.

3. D. Conway, "A Formation Flying Testbed," Distributed Spacecraft Control Workshop I, December 1998.

4. M. E. Hametz, D. J. Conway, D. A. Weidow, "New Tools for Formation Monitoring and Maintenance," 1999 American Control Conference, San Diego, CA, June 1999.

MISSION PERFORMANCE MEASURES FOR SPACECRAFT FORMATION FLYING

Steven P. Hughes and Christopher D. Hall
Aerospace and Ocean Engineering
Virginia Polytechnic Institute and State University

ABSTRACT

Clusters of low-performance spacecraft flying in formation may provide enhanced performance over single high-performance spacecraft. This is especially true for remote sensing missions where interferometry or stereographic imaging may provide higher resolution data. The configuration of such formations vary during an orbit due to orbital dynamics, and over larger time scales due to perturbations. Selection of a configuration should be based on overall performance of the formation. In this paper, performance measures are developed and evaluated based on integration over one orbit. The measures involve angular separation of spacecraft, instantaneous overlap access area, and an area-based measure of the separation of the spacecraft. An optimization scheme is used to determine the best configuration for a four-spacecraft formation.

INTRODUCTION

The formation flying concept has recently become popular due to its advantages in remote sensing missions and flexible long term mission capabilities. A multiple spacecraft platform allows increased flexibility in mission scenarios as new technology is developed. Distribution of components on a number of satellites allows the advantage that a single component failure results in the replacement of a small, cheap spacecraft and not mission failure. Areas of formation flying application in remote sensing include stereographic viewing, interferometry, and synthetic apertures. The advantage of formation flying for remote sensing missions is primarily the difference in perspective obtained from different satellites in a formation. Therefore it is necessary to design the orbits of spacecraft in a formation so that there is an appropriate separation between them to provide different viewing perspectives of targets of interest.

Formation flying is a relatively new concept. Preliminary investigations into formation flying were considered by Folta, Bordi, and Scolese.[1] Relative navigation control algorithms have been developed by Folta and Quinn[2] and DeCou.[3] The feasibility of using Goddard Space Flight Center's (GSFC) Onboard Navigation System for relative navigation was examined by Gramling *et al.*[4] Wickert *et al.*[5] investigated the feasibility of using a distributed architecture for a space-based radar detection system. However, performance measures that allow comparative formation orbit design have not received significant attention in the literature.

We develop general formation performance measures similar to those for single satellite missions. Orbit performance measures are developed that evaluate the separation between spacecraft in a formation. Algorithms for calculating the suggested measures are developed. Finally we postulate a hypothetical formation and apply the performance measures to demonstrate their application and nature.

SYMBOLS

A_o	overlap area, m^2
A_{refo}	reference overlap area, m^2
A_s	separation area, m^2
A_{refs}	reference separation area, m^2
A_{inner}	inner overlap area, m^2
A_{outer}	outer overlap area, m^2
α_{ij}	angle between the i^{th} and j^{th} satellites, rad
β	angle between sub-satellite points, rad
R_e	Earth radius, m
n	number of spacecraft in formation
n_i	number of points defining overlap region
ρ,ϵ	angular radii of horizon circles, rad
T	orbital period, s
θ	rotation angle of inner overlap area, rad

FORMATION MODELING

Spacecraft are subject to disturbing forces such as atmospheric drag, solar radiation pressure, and third body effects as well as the non-spherical central body force. The orbit performance measures to be developed subsequently require knowledge of the position of all spacecraft in the formation over a period. For a comparative study of candidate orbits the performance measure must be evaluated for many different initial states, resulting in a large number of orbit integrations. High fidelity force models slow the performance but are not expected to have a significant effect on preliminary formation design. Therefore, to simplify the preliminary analysis we neglect the disturbing forces and approximate the orbital dynamics with Keplerian motion.

For a formation composed of n satellites there are $6n$ degrees of freedom. A large number of state variables makes it difficult to examine the change in a performance measure due to changes in the initial state. We explore the design space of a formation by decreasing the number of state variables used to define a formation uniquely.

To ensure the formation's periodic motion over an orbit we assume all spacecraft have the same semi-major axis. This results in all of the spacecraft having the same period because the period depends only on the orbital parameter and the semi-major axis. If all the spacecraft in a formation do not have the same period then eventually it will separate and there will be no common coverage area. If we hold the semi-major axis constant for all satellites we decrease the number of state variables from $6n$ to $5n$.

To further reduce the number of state variables we assume all spacecraft to be in circular orbits. We obviously lose n state variables by fixing the eccentricity to zero. We lose another n variables because it is now simpler to locate a satellite in its orbital plane. For non-circular orbits we require two orbital elements to locate a satellite in the orbital plane, the argument of periapsis and the true anomaly. However for orbits with zero eccentricity the argument of periapsis is undefined. For a circular orbit, to locate the spacecraft's position in the orbital plane we require only a single variable, the true longitude at epoch. We have now reduced the number of state variables to define a formation to $3n$.

PERFORMANCE MEASURES

Larson and Wertz[6] present several classical performance measures used to evaluate the effectiveness of orbit candidates. The instantaneous access area is commonly used to measure how much area a spacecraft is capable of viewing at a given instant. The area access rate is the rate at which land enters or leaves the access area. Multiple satellite missions are more complex and require more complicated performance measures.

Overlap Area

We define the overlap area as the area on the sphere that can be viewed simultaneously by all satellites in the formation as seen in Fig. 1. A measure of this type provides a conventional way of evaluating formation performance similar to the instantaneous access area for single satellite missions. The size of the overlap area is obviously dependent on the amount of separation between spacecraft in formation. For formations with small separations the overlap area is large and on the order of the instantaneous access area for a single satellite at the same radius. As formation separations increase the overlap area decreases.

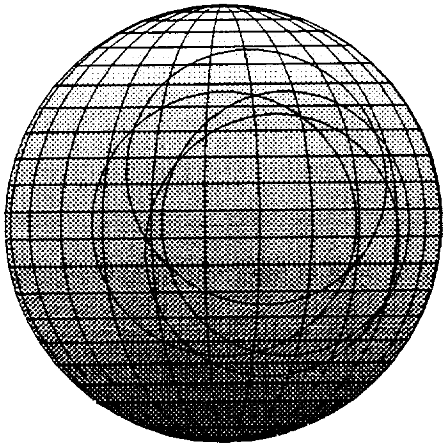


Figure 1: The Overlap Area

The algorithm to calculate the overlap area assumes a spherical Earth and uses spherical trigonometry[7] to calculate the area on the sphere. The overlap of small circles that define the horizons for each spacecraft defines the border of the region. For a point to lie in the overlap area it must either lie within the boundary of every horizon circle or on the borders. Using these criteria the intersection points defining the overlap region are determined. Knowing the defining intersection points the area is broken down into two subregions. The inner region is defined by connecting the intersection points with great circles. The outer region is defined by the lunes created by the horizon circles and the inner region. Calculation of the inner area is simple because its border is defined by great circles for which there is a simple spherical trigonometry relation:

$$A_{inner} = 2\pi R_e(1 - (n_i - 1))\theta \qquad (1)$$

Because the outer region is bounded by small circles, there is a considerable increase in the complexity of the area calculation:

$$A_{outer} \;=\; 2\pi R_e - 2R_e \cos\rho \arccos\left(\frac{\cos\epsilon - \cos\rho\cos\beta}{\sin\rho\sin\beta}\right) \qquad (2)$$

$$- 2R_e \cos\epsilon \arccos\left(\frac{\cos\rho - \cos\epsilon\cos\beta}{\sin\epsilon\sin\beta}\right)$$
$$- 2R_e \arccos\left(\frac{\cos\beta - \cos\epsilon\cos\rho}{\sin\epsilon\sin\rho}\right)$$

where ρ and ϵ are the angular radii of two small circles defining a lune and β is the center-to-center distance.

Separation Area

We define the separation area as the region enclosed by the outermost sub-satellite points as seen in Fig. 2. It is developed as a conflicting measure to the overlap area. For formations with small separations the separation area is small. As separations increase the separation area also increases. The border of the separation area is formed by connecting the outer sub-satellite points with great circles. Calculation of this area is simpler than the overlap area because its borders are defined by great circles. Once the border has been determined the rotation angle can be calculated using simple trigonometric relations and the area is calculated from Eq. (1).

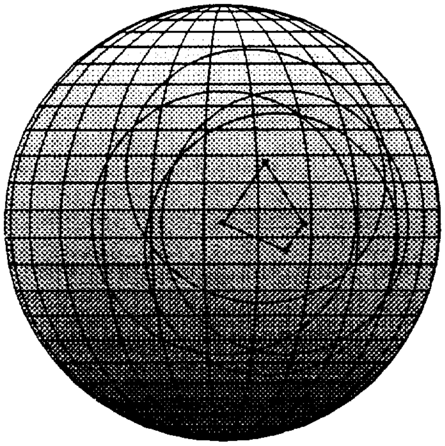


Figure 2: The Separation Area

Orbit Performance Measures

We desire measures that allow comparison of candidate orbits for satellites in formation. We develop a strategy that takes a measure of the instantaneous relative positions of a formation and integrate the measure over a period. The primary difference between the different orbit measures is the instantaneous relative position metric.

It is intuitive that for formations with little spacing the separation area is small while the overlap area is large. As separation of the formation increases the overlap area decreases and the separation area increases. The conflicting nature of the area measures can be utilized in an orbit performance measure such as,

$$W_1 = \frac{1}{T}\int_0^T \frac{1}{2}\left(\frac{A_s}{A_{refs}} + \frac{A_o}{A_{refo}}\right) dt \qquad (3)$$

This metric integrates the sum of the normalized area measures over a period. However, the calculation of the overlap area is expensive. The overlap region is of interest for mission performance but is too expensive to be used as a formation design tool. We wish to capture the nature of the area measure technique without the computational expense.

Angular Separation Metric

The measure involving overlap and separation areas is computationally expensive due to the spherical trigonometry relations required to calculate the overlap area. We desire a metric that penalizes the formation if it is too close or too separated without the numerical complexity of the area method. We use the fact that formations with large separation areas also have large angular separations between spacecraft. Similarly, formations with large overlap areas have small angular separations. A measure utilizing separation angles is much less computationally demanding because the angle between satellites is much simpler to calculate than the area measures. We formulate a new measure

$$W_2 = \frac{1}{T}\int_0^T \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w(\alpha_{ij})\, dt \tag{4}$$

The measure integrates over one period an instantaneous weight based on the angular separations of the satellites. The instantaneous weight w is a heuristic formula that measures the desirability of angular separation between the i^{th} and j^{th} spacecraft. Two forms for the function are considered as seen in Fig. 3. Both forms have a maximum of one at angular separations resulting in spatial separations of 3 km at a radius of 8000 km. The first function is a quartic polynomial which does not apply a negative weight for formations outside the allowed angular separation limit. The second function is parabolic and applies a negative weight for formations with unacceptable angular separations.

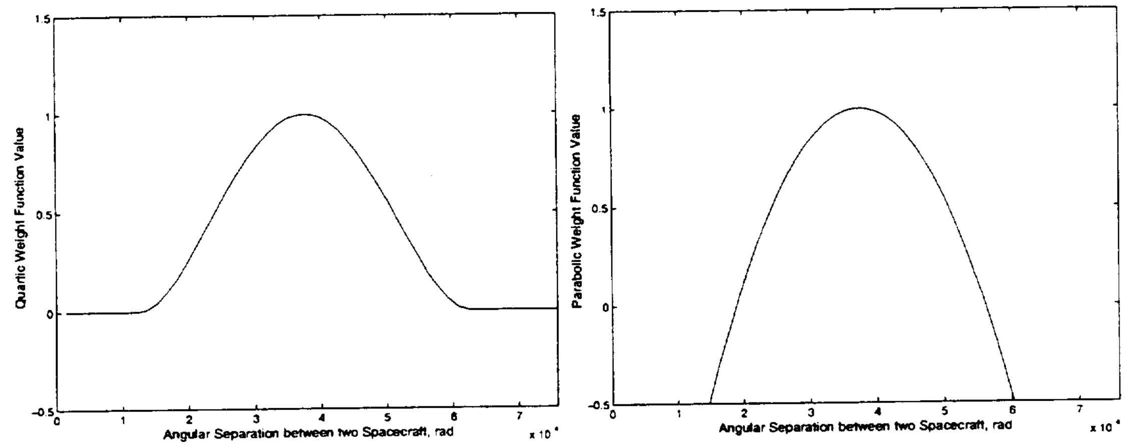


Figure 3: Heuristic Weight Functions, w

Weighted Separation Area Metric

We develop a third metric to capitalize on the relative simplicity of the separation area calculation in comparison to the overlap area. This measure is applicable to formations composed of more than 2 spacecraft because for 2 spacecraft scenarios the separation area is undefined. By applying a weight function to the separation area we penalize formations whose separation area is either too large or too small. This captures the conflicting nature of the overlap/separation measure with a significant decrease in computational complexity. By integrating the weight of the separation area over an orbit we can get a measure of orbit performance. Specifically

$$W_3 = \frac{1}{T} \int_0^T w(A_s)\, dt \tag{5}$$

For this case w is a heuristic formula of the form seen in Fig. 4. The ideal separation area occurs at $2 \times 10^6 m^2$. Allowable separations occur between $1 \times 10^6 m^2$ and $3 \times 10^6 m^2$. Configurations outside of this range receive a negative weight.

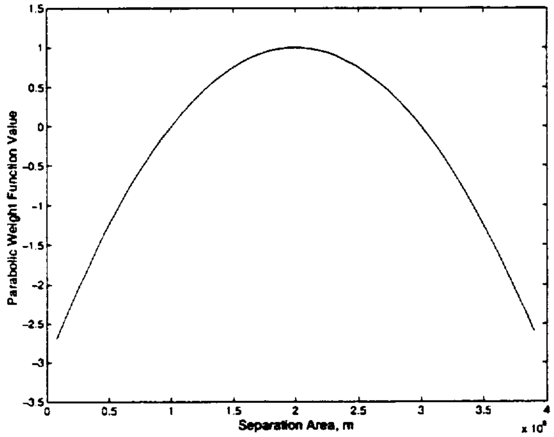


Figure 4: Parabolic Weight Function w for Angular Separation Metric

IMPLEMENTATION AND RESULTS

By fixing the semi-major axis for all spacecraft and assuming circular orbits we can represent a formation uniquely by $3n$ state variables. This is still a large number of state variables to examine the change in performance measures due to changes in the initial state. A further reduction in order is achieved by referencing all spacecraft to one of the spacecraft in the formation. Because the performance measures developed are not dependent on absolute position with respect to the Earth, only on the relative position of spacecraft, there is no loss of generality in doing so. We are simply picking one of a family of solutions that results in the same weight function value over an orbit. This is justified by realizing that there is an infinite number of state vectors such that the relative motion of the spacecraft is the same, only shifted in position with respect to the earth. Eliminating the three variables used to define the reference orbit we are left with $3(n-1)$ state variables to describe the formation.

As a preliminary example we define a formation to have four spacecraft with semi-major axes of 8000 km. We then have nine state variables to define the orbits uniquely. Although we have reduced the state variables from 24 to nine the resulting performance problem is still considerably complex. We now develop a representation that allows these nine variables to be represented by only two variables defined as the initial longitudinal and latitudinal separations. Since we are mainly concerned with the relative motion of the spacecraft, we have postulated a formation without concern to its path with respect to any specific points on the earth. We have fixed two spacecraft to be in circular, coplanar orbits. The separation between the coplanar spacecraft is defined as the longitudinal separation. The remaining two orbits are inclined at equal amounts with their ascending nodes placed 180° apart. At the initial epoch the satellites in the inclined orbits are at their maximum declination with respect to the coplanar orbits and lie midway between the two equatorial spacecraft. The initial separation between the inclined satellites is defined as the latitudinal separation.

Because we have only two state variables the effect of the initial state on the performance can be investigated using graphical techniques. The measures have been evaluated using MatLab over one period for a range of initial conditions. In all plots the independent variables have been converted from angular separation in radians, to spatial separation at the orbit radius to provide a more intuitive understanding.

Figure 5 shows the surface plot generated from the application of Eq. (4). To generate this plot the quartic angular weight function in Fig. 3 is used. The maximum value is seen to occur for a longitudinal separation of 4400 m and an initial latitudinal separation of 2800 m. The weight function is zero for all angles resulting in spatial separations of less than 1 km. As expected the performance measure is zero for those initial conditions that result in spatial separations always less than 1 km. Notice the longitudinal separation has a larger effect on the performance than the latitudinal separation. If the initial longitude is not acceptable according to the weight function it remains so over an orbit does and not contribute to the performance. The latitudinal separation has a different effect on performance due to the oscillatory nature of the inclined spacecraft with respect to the coplanar pair. If the initial separation is large and falls outside of the acceptable separation range there is still a contribution to the performance over the orbit. This is due to the fact that over an orbit the latitudinal separation starts at the initial value, goes to zero, and then increases back to original value. In doing so the angular separation between the inclined spacecraft must pass through the acceptable range and positively contribute to the performance.

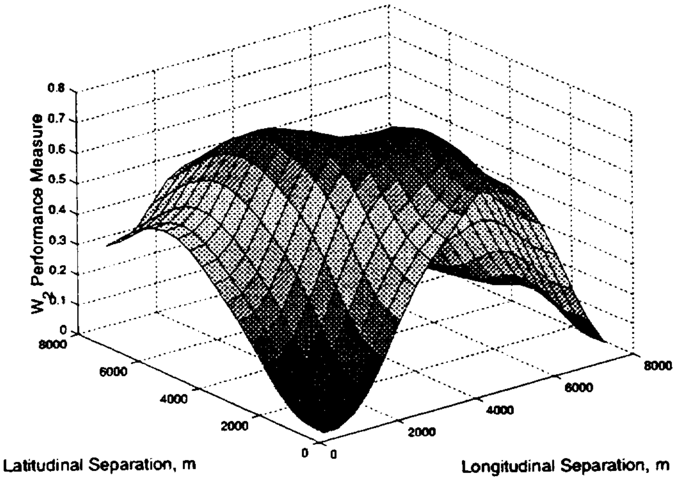


Figure 5: Angular Separation Measure with Quartic Weight Function

Application of the W_2 performance measure with the parabolic weight function is presented in Fig. 6. The optimum formation according to this performance measure is seen to occur for a longitudinal separation of 3190 m and an initial latitudinal separation of 3590 m. The effects of latitudinal and longitudinal separation on the parabolic case are similar to the quartic case. The longitudinal separation has a larger effect on the performance. The surface is smoother for the parabolic case than for the quartic weight function case. This smoothness is due primarily to the negative weight imposed by the parabolic function for separations that do not fall within the acceptable range.

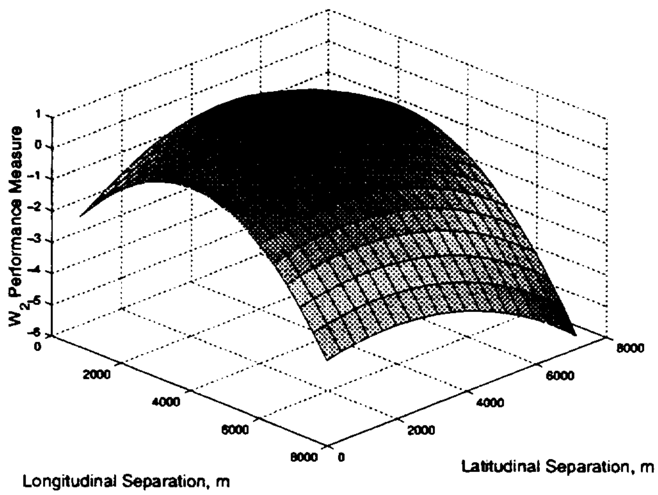


Figure 6: Angular Separation Measure with Parabolic Weight Function

The application of W_3 to the formation is seen in Fig. 7. The surface has a large region of near maximum weight as opposed to the surfaces in Fig. 5 and Fig. 6 that have distinct maxima. This demonstrates that there is a region within the design space that has near similar separation area integrals over an orbit. For scenarios with large longitudinal and latitudinal separations the performance is poor because the separation area is not acceptable over most of the orbit. If either of the separations goes to zero the separation area goes to zero and the performance measure tends toward the instantaneous weight function value at zero.

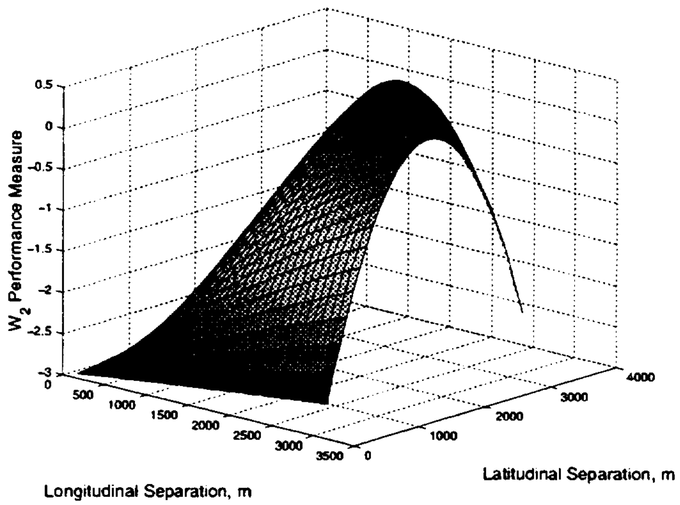


Figure 7: Weighted Separation Area Measure

We now investigate a higher order formation model using the W_2 performance measure with the parabolic instantaneous weight function seen in Fig. 3. Previously we justified representing the state of a formation of four spacecraft with nine variables. We now reduce the order by two to seven. Specifically we allow the longitude of the ascending node, inclination, and true longitude at epoch for the two inclined orbits to vary. We also allow the longitudinal separation to vary by allowing the true longitude at epoch of the leading coplanar spacecraft to vary.

For seven state variables the solution can no longer be found using graphical techniques. Instead we use Newton's method[8] to determine a maximum in the performance. To begin the iterative scheme we use the seven-variable representation of the optimal two-variable solution as an initial guess. The state vector is updated according to

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \left[\frac{\partial \mathbf{F}}{\partial \mathbf{x}}(\mathbf{x}_i)\right]^{-1} \mathbf{F}(\mathbf{x}_i) \tag{6}$$

where $\mathbf{F}(\mathbf{x}_n)$ is the gradient of W_2 and $\partial\mathbf{F}(\mathbf{x}_n)/\partial\mathbf{x}$ is the Jacobian of W_2. Because analytical derivatives of W_2 are not available approximations for the gradient and the Jacobian are found using one-sided finite differencing.

A maximum in the performance measure W_2 using the parabolic weight function is found. To illustrate the results the relative motion of the spacecraft has been exaggerated and illustrated in Fig. 8. There is no relative motion between the coplanar spacecraft so they are seen as fixed points. The inclined orbits produce figure eight patterns over one orbit. For clarity we label the left figure eight as orbit one and the right as orbit two. At the initial epoch the spacecraft in orbit one is at the top of the figure eight. We define clockwise motion for a figure eight to be the type of motion occurring in the upper half of the figure. Spacecraft one moves in clockwise motion according to this definition. The spacecraft in orbit two begins at the bottom of the figure eight and also moves in clockwise motion. For the actual example formation, these figure eight's appear as nearly straight vertical lines with a slight separation between them.

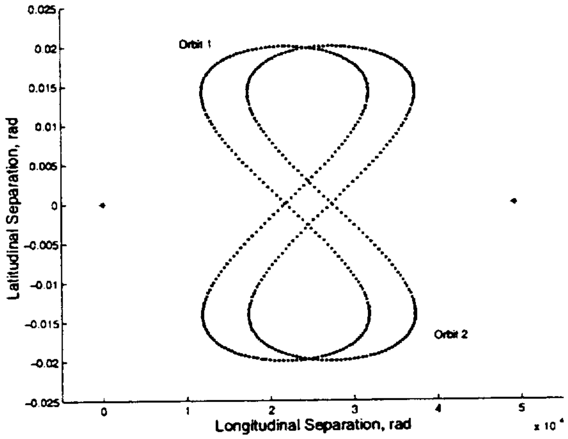


Figure 8: Exaggerated Relative Motion for Seven-Variable Solution

CONCLUSIONS

The goal of this paper is to contribute to the area of performance measures for formation flying satellites. We developed general measures similar to those for single spacecraft missions. The overlap area and the separation area represent conflicting measures of the amount of separation in a formation. Orbit performance measures were developed to ascertain if the separation in a formation is meeting necessary requirements over an orbit. The metrics are based on angular separation and area based measures of separation. Two formation models were postulated. A simple two degree of freedom formation was investigated and optimum configurations determined according to the performance measures using graphical techniques. A more complex seven-variable model was examined for one of the orbit performance metrics and an optimum solution found using Newton's method.

REFERENCES

[1] Folta, D., Bordi, F., and Scolese, S., "Considerations on Formation Flying Separations for Earth Observing Satellite Missions," *Advances in Astronautical Sciences*, Vol. 79, No. 2, 1992, pp. 803-822

[2] Folta, D., and Quinn, D., "Enhanced Formation Flying for the Earth Observing-1 (EO-1) New Millennium Mission," *NASA Conference Publication*, Vol. 3345, 1997, pp. 405-421

[3] DeCou, B.D., "Orbital Station-Keeping for Multiple Spacecraft Interferometry," *Journal of the Astronautical Sciences*, Vol. 39, No. 3, 1991, pp. 283-297

[4] Gramling, C.J., Lee, T., Niklewski, D.J., and Long, A.C., "Relative Navigation for Autonomous Formation Fling of Spacecraft," *Procedings of the 1997 AAS/AIAA Astrodynamics Specialist Conference*, AAS 97-627

[5] Wickert, D.P., Shaw, G.B., and Hastings, D., "Impact of a Distributed Architecture for Space-Based Radar," *Journal of Spacecraft and Rockets*, Vol. 35, No. 5, 1992, pp. 703-713

[6] Larson, W.J. and Wertz, J.R., *Space Mission Analysis and Design*, Microcosm, Inc., Torrance, 1992, Chaps. 5, 7

[7] Wertz, J.R (editor), *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, Dordrecht, 1997, Appendix A

[8] Dennis, Jr., J. E. and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

ATTITUDE TRACKING CONTROL FOR SPACECRAFT FORMATION FLYING

Matthew R. Long and Christopher D. Hall
Aerospace and Ocean Engineering
Virginia Polytechnic Institute and State University

ABSTRACT

We develop a non-linear tracking control law to be applied to formation flying spacecraft. Each spacecraft in the formation is modeled as a rigid body with N axisymmetric wheels controlled by axial torques, and the kinematics are represented by Modified Rodriques Parameters (MRPs). The paper first derives the open-loop reference attitude, rate, and acceleration commands for tracking a moving object with the sensor boresight vector defined along a body-fixed axis. The reference trajectory is constructed so that the solar panel normal is aligned with the sun vector at all times while tracking targets on the rotating earth. The controller makes the body frame asymptotically track the reference motion when there are initial errors in the position and angular rates. A simple target tracking example is presented to demonstrate that the controller will allow each spacecraft in the formation to track the target and the sun simultaneously.

INTRODUCTION

The formation flying concept has become a topic of interest in recent years. Gramling et al.[1] discussed the On-board Navigation System (ONS) for relative navigation of the Earth-Observing-1 (EOS-1)/Landsat-7 (L-7) formation. The performance of the ONS was investigated in terms of tracking measurement type and quality, tracking frequency, and the relative orbital geometry of the formation. DeCou[2] presented a station-keeping strategy for formation flying interferometry. He discussed the basic orbital configuration for interferometry missions and the thrust requirements for station-keeping of a two-satellite formation. The work done by Ulybyshev[3] pertains to station-keeping of a constellation using a linear-quadratic regulator for feedback control. The controller minimized the along-track relative displacements between spacecraft and the orbital period displacements relative to a reference orbit. Folta et al.[4] also addressed separations between spacecraft in a formation. The performance of a formation to observe ground targets simultaneously for various separations was evaluated. Simulation results for three different types of formations were presented in terms of attitude and field of view (FOV) errors.

Spacecraft rotational tracking maneuvers specifically for formation flying have not been addressed in the literature. The problem of tracking moving objects applicable to formation flying has been studied by various authors, and much of the work developed in this paper is based on Refs. 5, 6 and 7. Schaub et al.[8] also discussed rotational tracking maneuvers similar to what we present here, except that they optimized the reference trajectory for time and fuel requirements and used a different Lyapunov function to derive the momentum wheel controller. Steyn[9] and Wie and Lu[10] both investigated momentum wheel feedback controllers for rotational maneuvers. Slew rate constraints and near-minimum-time maneuvers were taken into account.

To determine the feasibility of formation simultaneous target tracking, we first consider the pointing and tracking requirements for an individual spacecraft. The desired attitude is constructed by making the sensor boresight axis co-linear with the position vector from the spacecraft to any arbitrary target. We define the target to be a point on the rotating earth, but it could be any inertially fixed, or moving target. We define two intermediate coordinate frames using the boresight axis, the solar array axis, and the sun vector to construct basis vectors that simultaneously allow the spacecraft to point at the target and keep the solar panel vector normal to the sun direction. The ideal tracking body rates and accelerations are computed from the first and second derivatives of the attitude, respectively. The reference acceleration is used to compute the ideal axial control torque for the control law. The controller uses Lyapunov control theory to drive any initial errors in the attitude and angular velocity to zero asymptotically. The

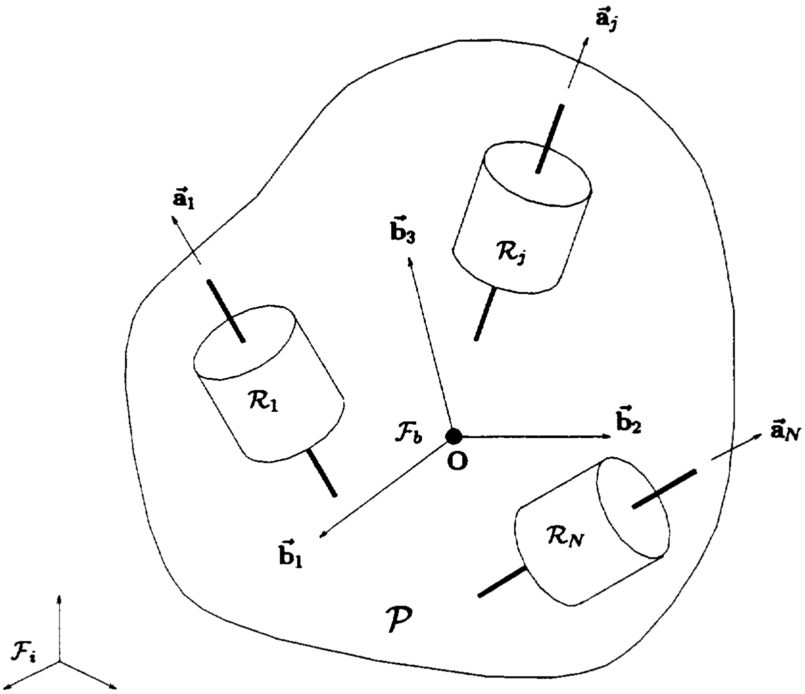


Figure 1: Rigid body with N momentum wheels

controller calculates the necessary control torque to track the specified trajectory, while countering all effects due to gravity gradient torques.

The first part of this paper defines the equations of motion an individual formation flying spacecraft model. The dynamics are presented, along with the kinematics expressed in terms of the body and reference frames. The second part of the paper outlines the open-loop computation of the ideal target tracking trajectory while keeping perfect alignment between the solar panel normal and the sun. The last part of the paper deals with the nonlinear feedback controller that makes the spacecraft body frame track the reference trajectory.

EQUATIONS OF MOTION

In this section, a system model is presented for use in developing tracking control algorithms. The equations of motion presented here follow the notation developed in Hughes[11] and Hall.[12] We first consider a rigid spacecraft P, shown in Figure 1, with N rigid momentum wheels W_i, $i = 1, \cdots, N$. The wheels have an arbitrary, but fixed orientation with respect to the body. Let $\mathcal{F}_b$ denote the body frame with the origin at the center of mass of the system $P + \sum_{i=1}^{N} W_i$, and $\mathcal{F}_i$ denote the inertial frame. The desired trajectory to be tracked comes from the trajectory generated by a "virtual" spacecraft in a reference frame. Let $\mathcal{F}_r$ represent this reference frame which is fixed at the center of mass of this virtual spacecraft. In Ref. 7, the "virtual" spacecraft is assumed to be a rigid body. Here we assume that the virtual spacecraft is a rigid body with momentum wheels, *i.e.* a gyrostat, with the same properties as the real spacecraft.

Let $\mathbf{I}$ represent the moment of inertia of the system, including the momentum wheels, and $\mathbf{I}_s = \text{diag}\{I_{s1}, \cdot, I_{sN}\}$ denote the axial moments of inertia of each momentum wheel. The dynamics of the gyrostat are given by

$$\dot{\mathbf{h}}_b = \mathbf{h}_b^\times \mathbf{J}^{-1}(\mathbf{h}_b - \mathbf{A}\mathbf{h}_a) + \mathbf{g}_e \qquad (1)$$

$$\dot{\mathbf{h}}_a = \mathbf{g}_a \qquad (2)$$

$$\dot{\boldsymbol{\sigma}}_b = \mathbf{G}(\boldsymbol{\sigma}_b)\boldsymbol{\omega}_b \tag{3}$$

where "×" denotes a skew-symmetric matrix, $\mathbf{h}_b$ is the 3×1 system angular momentum vector in $\mathcal{F}_b$

$$\mathbf{h}_b = \mathbf{I}\boldsymbol{\omega}_b + \mathbf{A}\mathbf{I}_s\boldsymbol{\omega}_s \tag{4}$$

and $\mathbf{h}_a$ is the $N \times 1$ matrix of the axial angular momenta of the wheels defined as

$$\mathbf{h}_a = \mathbf{I}_s\mathbf{A}^T\boldsymbol{\omega}_b + \mathbf{I}_s\boldsymbol{\omega}_s \tag{5}$$

Here $\boldsymbol{\omega}_s$ is an $N \times 1$ matrix that describes the axial angular velocities of the momentum wheels relative to the body. The $3 \times N$ matrix $\mathbf{A}$ contains the axial unit vectors of the N momentum wheels, and $\mathbf{J}$ is an inertia-like matrix[12] defined as

$$\mathbf{J} = \mathbf{I} - \mathbf{A}\mathbf{I}_s\mathbf{A}^T \tag{6}$$

From Eqs. (4), (5), and (6) it can be shown that the angular velocity of the body frame can be written as

$$\boldsymbol{\omega}_b = \mathbf{J}^{-1}(\mathbf{h}_b - \mathbf{A}\mathbf{h}_a) \tag{7}$$

The term $\mathbf{g}_a$ represents the $N \times 1$ matrix of the internal axial torques applied by the platform to the momentum wheels. This is the control torque needed for tracking manuevers. In this paper, the only external torque that our controller compensates for is the uncontrolled gravity gradient torque[13] $\mathbf{g}_e$:

$$\mathbf{g}_e = 3\,\frac{\mu}{\|\mathbf{r}_{so}\|^3}\,\hat{\mathbf{o}}_3^{\times}\mathbf{I}\hat{\mathbf{o}}_3 \tag{8}$$

where $\hat{\mathbf{o}}_3$ is the nadir vector, *i.e.* $\hat{\mathbf{o}}_3 = -\mathbf{r}_{so}/\|\mathbf{r}_{so}\|$, with $\mathbf{r}_{so}$ being the position vector from the center of the earth to the center of mass of the spacecraft in the orbital frame $\mathcal{F}_o$. The kinematics in Eq. (3) are written in terms of Modified Rodriques Parameters (MRP's). The MRP's are a three-parameter set derived from the Euler axis/angle representation[14] and are defined:

$$\boldsymbol{\sigma} = \hat{\mathbf{e}}\tan(\Phi/4) \tag{9}$$

where $\hat{\mathbf{e}}$ is the unit vector along the Euler principal axis, and Φ is the Euler principal rotation angle. The matrix $\mathbf{G}(\boldsymbol{\sigma}_b)$ in Eq. (3) is defined as

$$\mathbf{G}(\boldsymbol{\sigma}_b) = \frac{1}{2}\left(\mathbf{1} + \boldsymbol{\sigma}_b^{\times} + \boldsymbol{\sigma}_b^T\boldsymbol{\sigma}_b - \frac{1+\boldsymbol{\sigma}_b^T\boldsymbol{\sigma}_b}{2}\mathbf{1}\right) \tag{10}$$

where $\mathbf{1}$ is the 3×3 identity matrix.

Since the virtual spacecraft has the same inertial and wheel parameters as the real spacecraft, the reference frame dynamics are the same as Eqs. (1–5), except that the subscript b is replaced with r. The uncontrolled external torque $\mathbf{g}_e$ remains the same and the reference wheel torque $\mathbf{g}_{ar}$ comes from first noting that $\dot{\mathbf{h}}_r$ can also be expressed as

$$\dot{\mathbf{h}}_r = \mathbf{J}\dot{\boldsymbol{\omega}}_r + \mathbf{A}\dot{\mathbf{h}}_{ar} = \mathbf{J}\dot{\boldsymbol{\omega}}_r + \mathbf{A}\mathbf{g}_{ar} \tag{11}$$

Equating Eqs. (11) and (1) in terms of $\mathcal{F}_r$ yields the following expression for the desired axial control torque $\mathbf{A}\mathbf{g}_{ar}$:

$$\mathbf{A}\mathbf{g}_{ar} = \mathbf{h}_r^{\times}\mathbf{J}^{-1}(\mathbf{h}_r - \mathbf{A}\mathbf{h}_{ar}) + \mathbf{g}_e - \mathbf{J}\dot{\boldsymbol{\omega}}_r \tag{12}$$

where $\mathbf{J}^{-1}(\mathbf{h}_r - \mathbf{A}\mathbf{h}_{ar})$ is the desired angular velocity $\boldsymbol{\omega}_r$. The torque $\mathbf{g}_{ar}$ is the torque that would generate the desired trajectory without any initial condition errors. Likewise with the kinematics, the reference MRP's are given by Eq. (3) with the subscript r instead of b.

TRACKING KINEMATICS

Tracking a moving object involves two kinematical aspects: pointing at an object and then moving at the correct rate to stay aligned with the target for a given length of time. In this section, we develop the ideal attitude, rate, and acceleration commands needed for target tracking. The desired trajectory is computed in an open-loop fashion, and is defined to be the trajectory of the virtual spacecraft in $\mathcal{F}_r$. The controller will then make the body frame track this reference motion asymptotically.

Pointing

Pointing at a target requires a specific attitude to make the position vector from the target to the spacecraft co-linear with the instrument boresight as illustrated in Fig. 2. We define the target to be a point on the rotating earth defined by its latitude and longitude. We also require that the attitude be constructed so that the sun unit vector $\hat{\mathbf{s}}$ is perpendicular to the solar panel unit vector $\hat{\mathbf{p}}$ while pointing at the target. This sun tracking is also known as yaw-steering.[15]

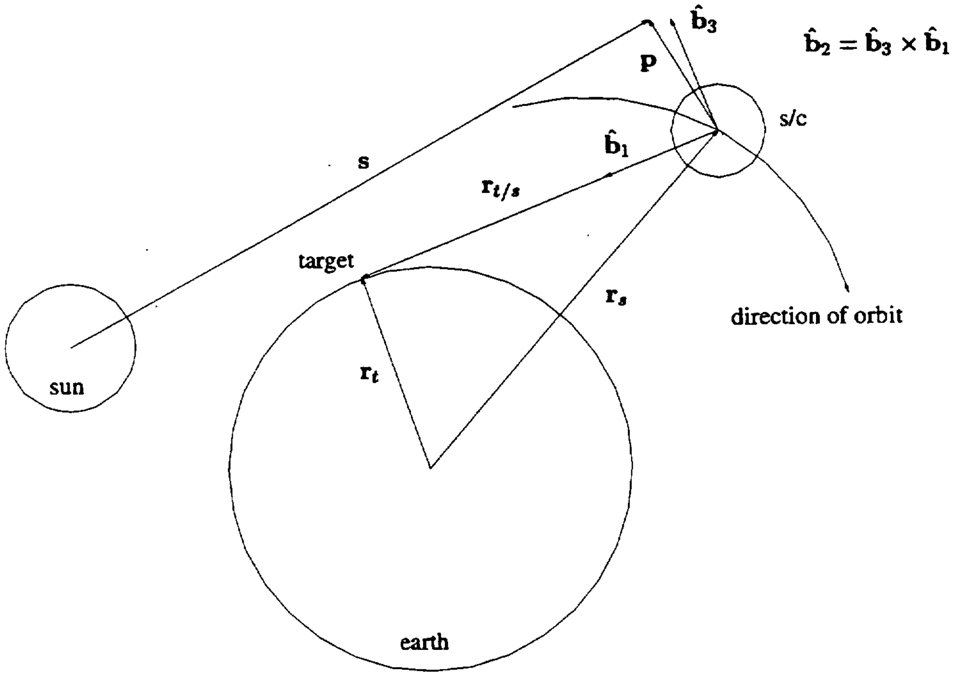


Figure 2: Setup for Target Tracking.

The instrument axis can be any unit vector fixed in $\mathcal{F}_b$, which is defined to be the same in $\mathcal{F}_r$. In this paper, we assume that the instrument boresight is defined along the "1" direction in $\mathcal{F}_b$ and $\mathcal{F}_r$:

$$\mathbf{a}_b = \mathbf{a}_r = [1\ \ 0\ \ 0]^T \tag{13}$$

From Figure 2, it can easily be seen that the target position vector with respect to the spacecraft, $\mathbf{r}_{t/s}$, can be written in $\mathcal{F}_i$ as

$$\mathbf{r}_{t/si} = \mathbf{r}_{ti} - \mathbf{r}_{si} \tag{14}$$

where $\mathbf{r}_{ti}$ represents the position vector from the center of the earth to the spacecraft in $\mathcal{F}_i$. The vector $\mathbf{r}_{si}$ is assumed known, and the target position vector is expressed in $\mathcal{F}_i$ as

$$\mathbf{r}_{ti} = [\cos(\delta_t)\cos(\theta_{GST} + L_t)\ \ \cos(\delta_t)\sin(\theta_{GST} + L_t)\ \ \sin(\delta_t)]^T \tag{15}$$

where δ_t and L_t are the latitude and longitude of target, respectively, and θ_{GST} is Greenwich sidereal time measured from a given epoch.

To find the required pointing attitude, we develop the rotation matrix $\mathbf{R}^{ri}$ that will make the condition in Fig. (2) true and satisfy the yaw-steering condition:

$$\hat{\mathbf{s}} \cdot \hat{\mathbf{p}}_r = 0 \tag{16}$$

We define two intermediate frames $\mathcal{F}_a$ and $\mathcal{F}_c$ that are used with the frames $\mathcal{F}_o$ and $\mathcal{F}_r$ to construct the rotation matrix $\mathbf{R}^{ri}$. Our motivation for this approach is that given the position vectors in Figure 2, $\mathbf{a}$, $\hat{\mathbf{s}}$, and $\hat{\mathbf{p}}$, the attitude that will align the boresight with the target and the solar panel normal with the sun can be constructed at once by the following:

$$\mathbf{R}^{ri} = \mathbf{R}^{rc}\mathbf{R}^{ca}\mathbf{R}^{ao}\mathbf{R}^{oi} \tag{17}$$

An alternative method is to develop the necessary pointing attitude without the yaw steering, and then determine the rotation about the sensor axis needed to satisfy Eq. (16). This approach is less attractive since it requires an extra computation in the algorithm.

The rotation matrix $\mathbf{R}^{oi}$ is determined by the spacecraft inertial position and velocity vectors $\mathbf{r}_{si}$ and $\mathbf{v}_{si}$, respectively, and is given by the following

$$\hat{\mathbf{o}}_{3i} = \frac{-\mathbf{r}_{si}}{||\mathbf{r}_{si}||} \tag{18}$$

$$\hat{\mathbf{o}}_{2i} = \frac{\mathbf{r}_{si}^{\times}\mathbf{v}_{si}}{||\mathbf{r}_{si}^{\times}\mathbf{v}_{si}||} \tag{19}$$

$$\hat{\mathbf{o}}_{1i} = \frac{\hat{\mathbf{o}}_{2i}^{\times}\hat{\mathbf{o}}_{3i}}{||\hat{\mathbf{o}}_{2i}^{\times}\hat{\mathbf{o}}_{3i}||} \tag{20}$$

$$\mathbf{R}^{oi} = [\hat{\mathbf{o}}_{1i}\ \hat{\mathbf{o}}_{2i}\ \hat{\mathbf{o}}_{3i}]^T \tag{21}$$

We construct the basis of $\mathcal{F}_a$ using the boresight axis $\mathbf{a}$ and the sun vector $\hat{\mathbf{s}}$. These vectors, known in the orbital frame using Eq. (21), are used to contruct the rotation matrix $\mathbf{R}^{ao}$. As with $\mathbf{R}^{oi}$, the rotation matrix between $\mathcal{F}_o$ and $\mathcal{F}_a$ is

$$\hat{\mathbf{a}}_{1o} = \mathbf{a}_o \tag{22}$$

$$\hat{\mathbf{a}}_{3o} = \frac{\mathbf{a}_o^{\times}\hat{\mathbf{s}}_o}{||\mathbf{a}_o^{\times}\hat{\mathbf{s}}_o||} \tag{23}$$

$$\hat{\mathbf{a}}_{2o} = \hat{\mathbf{a}}_{3o}^{\times}\hat{\mathbf{a}}_{1o} \tag{24}$$

$$\mathbf{R}^{ao} = [\hat{\mathbf{a}}_{1o}\ \hat{\mathbf{a}}_{2o}\ \hat{\mathbf{a}}_{3o}]^T \tag{25}$$

The basis of $\mathcal{F}_c$ is defined by the solar panel axis $\hat{\mathbf{p}}$, and the boresight axis vector $\mathbf{a}$, both of which are known in $\mathcal{F}_b$ and hence, $\mathcal{F}_r$. As before, the rotation matrix $\mathbf{R}^{rc}$ is constructed by

$$\hat{\mathbf{c}}_{1r} = \mathbf{a}_r \tag{26}$$

$$\hat{\mathbf{c}}_{3r} = \frac{\mathbf{a}_b^{\times}\hat{\mathbf{p}}_r}{||\mathbf{a}_r^{\times}\hat{\mathbf{p}}_r||} \tag{27}$$

$$\hat{\mathbf{c}}_{2r} = \hat{\mathbf{c}}_{3r}^{\times}\hat{\mathbf{c}}_{1r} \tag{28}$$

$$\mathbf{R}^{rc} = [\hat{\mathbf{c}}_{1r}\ \hat{\mathbf{c}}_{2r}\ \hat{\mathbf{c}}_{3r}]^T \tag{29}$$

To determine the rotation matrix from $\mathcal{F}_c$ to $\mathcal{F}_a$, we use the prescribed orthogonality condition between the sun vector and the solar panel axis. Equation (16) can be expressed as

$$\hat{\mathbf{s}}_a^T\mathbf{R}^{ac}\hat{\mathbf{p}}_c = 0 \tag{30}$$

From Eqs. (22) and (26), it is easy to see that $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{c}}_1$ are the same vector, therefore $\hat{\mathbf{a}}_1 \cdot \hat{\mathbf{c}}_1 = 1$. By definition, the unit vectors $\hat{\mathbf{a}}_2$ and $\hat{\mathbf{a}}_3$ are perpendicular to $\hat{\mathbf{a}}_1$, so they are also perpendicular to $\hat{\mathbf{c}}_1$. The same is true for $\hat{\mathbf{a}}_1$, which is also perpendicular to $\hat{\mathbf{c}}_2$ and $\hat{\mathbf{c}}_3$. As a result, $\mathbf{R}^{ac}$ is a "1" rotation and Eq. (30) becomes

$$\hat{\mathbf{s}}_a^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{ac} & \sin\theta_{ac} \\ 0 & -\sin\theta_{ac} & \cos\theta_{ac} \end{bmatrix} \hat{\mathbf{p}}_c = 0 \tag{31}$$

Because of the way we have defined $\mathcal{F}_a$ and $\mathcal{F}_c$, s_{3a} and p_{3c} are zero and Eq. (31) expands as

$$s_{1a}p_{1c} + s_{2a}p_{2c}\cos\theta_{ac} = 0 \tag{32}$$

The angle θ_{ac} is then easily found by solving Eq. (32), which satisfies the yaw-steering condition. $\mathbf{R}^{ac}$ is then calculated by substituting θ_{ac} into Eq. (31). The ideal target pointing attitude $\mathbf{R}^{ri}$ is then constructed by multiplying together the rotation matrices found in Eq. (17).

Tracking

To keep the spacecraft pointed at the target, the spacecraft must rotate as it moves in its orbit. We develop rate and acceleration commands similar to those found in Ref. 6. We begin by taking a time derivative of Eq. (14) to get

$$\dot{\mathbf{r}}_{t/si} = \dot{\mathbf{r}}_{ti} - \dot{\mathbf{r}}_{si} \tag{33}$$

where $\dot{\mathbf{r}}_{si}$ is simply the known spacecraft velocity vector and $\dot{\mathbf{r}}_{ti}$ is given by $\boldsymbol{\omega}_e^{\times}\mathbf{r}_{ti}$, where $\boldsymbol{\omega}_e$ is the angular velocity of the earth.

For the spacecraft to track a given target correctly, the angular velocity has to be coupled to the attitude. We preserve this coupling by calculating the spacecraft angular rate vectors in each of the coordinate frames and then use the pieces to construct the correct angular tracking rate in the body frame, which is

$$\boldsymbol{\omega}_r^{ri} = \boldsymbol{\omega}_r^{ca} + \boldsymbol{\omega}_r^{ai} \tag{34}$$

This is analogous to the calculation of each rotation matrix to get from $\mathcal{F}_i$ to $\mathcal{F}_r$ in the previous section.

We begin by defining the orbital rate vector in $\mathcal{F}_o$. Since we are assuming that the orbit is circular, the orbital rate vector is just the mean motion of the orbit expressed as

$$\boldsymbol{\omega}_o^{oi} = \begin{bmatrix} 0 & -\sqrt{\mu/\|\mathbf{r}_{si}^3\|} & 0 \end{bmatrix}^T \tag{35}$$

The superscript oi denotes the angular velocity of $\mathcal{F}_o$ with respect to $\mathcal{F}_i$, the subscript o shows that the rate vector is expressed in $\mathcal{F}_o$, and μ is the earth's gravitational parameter.

Since $\mathcal{F}_a$ and $\mathcal{F}_o$ change with time, we need to calculate the angular velocity $\boldsymbol{\omega}_a^{ao}$. It is not difficult to show[11] that $\boldsymbol{\omega}_a^{ao}$ can be calculated based on the differentiation of $\mathbf{R}^{ao}$ as

$$\boldsymbol{\omega}_a^{ao\times} = -\overset{\circ}{\mathbf{R}}{}^{ao}\mathbf{R}^{oa} \tag{36}$$

where "o" denotes differentiation with respect to a moving coordinate frame. The matrix $\overset{\circ}{\mathbf{R}}{}^{ao}$ is found by first rewriting Eqs. (22)–(24) as

$$\begin{aligned} D_1\hat{\mathbf{a}}_{1o} &= \mathbf{r}_{t/so} && (37)\\ D_2\hat{\mathbf{a}}_{3o} &= \mathbf{r}_{t/so}^{\times}\mathbf{s}_o && (38)\\ \hat{\mathbf{a}}_{2o} &= \hat{\mathbf{a}}_{3o}^{\times}\hat{\mathbf{a}}_{1o} && (39) \end{aligned}$$

where the scalar quantities D_1 and D_2 are given by $\|\mathbf{r}_{t/so}\|$ and $\|\mathbf{r}_{t/so}^{\times}\hat{\mathbf{s}}_o\|$, respectively. Differentiating the above equations with respect to time results in the following

$$\begin{aligned} \dot{\hat{\mathbf{a}}}_{1o} &= \frac{\dot{\mathbf{r}}_{t/so} - \dot{D}_1\hat{\mathbf{a}}_{1o}}{D_1} && (40)\\ \dot{\hat{\mathbf{a}}}_{3o} &= \frac{\dot{\mathbf{r}}_{t/so}^{\times}\hat{\mathbf{s}}_o - \dot{D}_2\hat{\mathbf{a}}_{3o}}{D_2} && (41)\\ \dot{\hat{\mathbf{a}}}_{2o} &= \dot{\hat{\mathbf{a}}}_{3o}^{\times}\hat{\mathbf{a}}_{1o} + \hat{\mathbf{a}}_{3o}^{\times}\dot{\hat{\mathbf{a}}}_{1o} && (42) \end{aligned}$$

where the above has been simplified by noticing that the sun vector slowly varies in the inertial frame, and can be assumed to be fixed throughout the spacecraft trajectory thus, eliminating $\dot{\hat{\mathbf{s}}}_o$. The rates of change of D_1 and D_2 are

$$\dot{D}_1 = \frac{\dot{\mathbf{r}}_{t/so} \cdot \mathbf{r}_{t/so}}{D_1} \tag{43}$$

$$\dot{D}_2 = \frac{(\dot{\mathbf{r}}^{\times}_{t/so}\hat{\mathbf{s}}_o) \cdot (\mathbf{r}^{\times}_{t/so}\hat{\mathbf{s}}_o)}{D_2} \tag{44}$$

These time derivatives have been computed with respect to $\mathcal{F}_i$. We need the time derivatives with respect to the moving orbital frame, which we find by making use of the following velocity equation from analytical dynamics[16]

$$\dot{\mathbf{v}} = \mathring{\mathbf{v}} + \boldsymbol{\omega}^{\times}\mathbf{v} \tag{45}$$

where $\mathbf{v}$ represents any vector expressed in a frame with angular velocity $\boldsymbol{\omega}$. We write the expressions for the unit vector rates of change with respect to $\mathcal{F}_o$ as

$$\mathring{\hat{\mathbf{a}}}_{1o} = \dot{\hat{\mathbf{a}}}_{1o} - \boldsymbol{\omega}^{oi\times}_{o}\mathbf{a}_{1o} \tag{46}$$

$$\mathring{\hat{\mathbf{a}}}_{3o} = \dot{\hat{\mathbf{a}}}_{3o} - \boldsymbol{\omega}^{oi\times}_{o}\mathbf{a}_{3o} \tag{47}$$

$$\mathring{\hat{\mathbf{a}}}_{2o} = \mathring{\hat{\mathbf{a}}}^{\times}_{3o}\,\hat{\mathbf{a}}_{1o} + \hat{\mathbf{a}}^{\times}_{3o}\,\mathring{\hat{\mathbf{a}}}_{1o} \tag{48}$$

and then the rate of change of $\mathbf{R}^{ao}$ with respect to $\mathcal{F}_o$ is simply

$$\mathring{\mathbf{R}}^{ao} = \begin{bmatrix} \mathring{\hat{\mathbf{a}}}_{1o} & \mathring{\hat{\mathbf{a}}}_{2o} & \mathring{\hat{\mathbf{a}}}_{3o} \end{bmatrix}^T \tag{49}$$

Using Eq. (36), the angular velocity $\boldsymbol{\omega}^{ai}_{a}$ is then found to be

$$\boldsymbol{\omega}^{ai}_{a} = \boldsymbol{\omega}^{ao}_{a} + \mathbf{R}^{ao}\boldsymbol{\omega}^{oi}_{o} \tag{50}$$

It can be seen from Eq. (31) that $\boldsymbol{\omega}^{ca}_{a}$ is simply

$$\boldsymbol{\omega}^{ca}_{a} = \begin{bmatrix} \dot{\theta}_{ac} & 0 & 0 \end{bmatrix}^T \tag{51}$$

and $\dot{\theta}_{ac}$ is found by taking a time derivative of Eq.(32) to yield

$$\dot{\theta}_{ac} = \frac{-\dot{s}_{1a}p_{1c} - \dot{s}_{2a}p_{2c}\cos\theta_{ac}}{s_{2a}p_{2c}\sin\theta_{ac}} \tag{52}$$

where the derivative of the sun vector with respect to $\mathcal{F}_a$ is given by

$$\mathring{\hat{\mathbf{s}}}_a = -\boldsymbol{\omega}^{ai\times}_{a}\hat{\mathbf{s}}_a \tag{53}$$

The frame $\mathcal{F}_c$ has a fixed orientation with respect to $\mathcal{F}_r$, so $\boldsymbol{\omega}^{rc}_{r} = \mathbf{0}$. As a result, the desired tracking body rate vector $\boldsymbol{\omega}^{ri}_{r}$ is constructed by adding Eqs. (50) and (51), and then rotating them into the reference frame to get

$$\boldsymbol{\omega}^{ri}_{r} = \mathbf{R}^{ra}(\boldsymbol{\omega}^{ca}_{a} + \boldsymbol{\omega}^{ai}_{a}) \tag{54}$$

where $\mathbf{R}^{ra}$ is the rotation matrix from $\mathcal{F}_a$ to $\mathcal{F}_r$ and is found from the previous section to be

$$\mathbf{R}^{ra} = \mathbf{R}^{rc}[\mathbf{R}^{ac}]^T \tag{55}$$

Once the angular rates are known in each of the coordinate frames, we compute the desired angular accelerations. The accelerations are needed to compute the reference axial wheel torque $\mathbf{g}_{ar}$ to generate the desired trajectory. The angular acceleration $\dot{\boldsymbol{\omega}}_r^{ri}$ is constructed analogously to the angular velocity $\boldsymbol{\omega}_r^{ri}$ by differentiating Eqs. (33–53). The acceleration commands are found by taking a time derivative of Eq. (33)

$$\ddot{\mathbf{r}}_{t/si} = \ddot{\mathbf{r}}_{ti} - \ddot{\mathbf{r}}_{si} \tag{56}$$

Here $\ddot{\mathbf{r}}_{si}$ is simply the two-body equation of motion[17] given as

$$\ddot{\mathbf{r}}_{si} = -\frac{\mu}{\|\mathbf{r}_{si}\|^3}\mathbf{r}_{si} \tag{57}$$

It can be shown from basic kinematics[18] and from Ref. 5 that the inertial acceleration of the target $\ddot{\mathbf{r}}_{ti}$ is

$$\ddot{\mathbf{r}}_{ti} = \frac{\omega_E^2 \mathbf{r}_{ti}}{\|\mathbf{r}_{ti}\|}\hat{\mathbf{n}}_t + \|\mathbf{r}_{ti}\|\omega_e^2\hat{\mathbf{u}}_t \tag{58}$$

where $\hat{\mathbf{n}}_t$ and $\hat{\mathbf{u}}_t$ are the normal and tangential unit vectors of the target motion.

The angular acceleration $\dot{\boldsymbol{\omega}}_o^{oi}$ is found by differentiating Eq. (35):

$$\dot{\boldsymbol{\omega}}_o^{oi} = \left[0 \;\; 1.5\mu/\left(\|\mathbf{r}_{si}^4\|\sqrt{\mu/\|\mathbf{r}_{si}^3\|}\right) \;\; 0\right]^T \tag{59}$$

The angular acceleration $\mathring{\boldsymbol{\omega}}_a^{ao}$ is found by differentiating Eq. (36) which yields

$$\mathring{\boldsymbol{\omega}}_a^{ao\times} = -\overset{\circ\circ}{\mathbf{R}}{}^{ao}\mathbf{R}^{oa} - \mathring{\mathbf{R}}^{ao}\mathring{\mathbf{R}}^{oa} \tag{60}$$

The second derivative of $\mathbf{R}^{ao}$ is found by differentiating Eqs. (46–48):

$$\begin{aligned}
\overset{\circ\circ}{\hat{\mathbf{a}}}_{1o} &= \ddot{\hat{\mathbf{a}}}_{1o} - 2\boldsymbol{\omega}_o^{oi\times}\dot{\mathbf{a}}_{1o} - \boldsymbol{\omega}_o^{oi\times}(\boldsymbol{\omega}_o^{oi\times}\mathbf{a}_{1o}) - \dot{\boldsymbol{\omega}}_o^{oi\times}\mathbf{a}_{1o} && (61)\\
\overset{\circ\circ}{\hat{\mathbf{a}}}_{3o} &= \ddot{\hat{\mathbf{a}}}_{3o} - 2\boldsymbol{\omega}_o^{oi\times}\dot{\mathbf{a}}_{3o} - \boldsymbol{\omega}_o^{oi\times}(\boldsymbol{\omega}_o^{oi\times}\mathbf{a}_{3o}) - \dot{\boldsymbol{\omega}}_o^{oi\times}\mathbf{a}_{3o} && (62)\\
\overset{\circ\circ}{\hat{\mathbf{a}}}_{2o} &= \overset{\circ\circ}{\hat{\mathbf{a}}}{}_{3o}^{\times}\,\hat{\mathbf{a}}_{1o} + 2\,\mathring{\hat{\mathbf{a}}}{}_{3o}^{\times}\,\mathring{\hat{\mathbf{a}}}_{1o} + \hat{\mathbf{a}}_{3o}^{\times}\overset{\circ\circ}{\hat{\mathbf{a}}}_{1o} && (63)\\
\overset{\circ\circ}{\mathbf{R}}{}^{ao} &= \begin{bmatrix}\overset{\circ\circ}{\hat{\mathbf{a}}}_{1o} & \overset{\circ\circ}{\hat{\mathbf{a}}}_{2o} & \overset{\circ\circ}{\hat{\mathbf{a}}}_{3o}\end{bmatrix}^T && (64)
\end{aligned}$$

where the inertial derivatives of the unit vectors are found by differentiating Eqs. (40–41):

$$\begin{aligned}
\ddot{\hat{\mathbf{a}}}_{1o} &= \frac{\ddot{\mathbf{r}}_{t/so} - 2\dot{D}_1\dot{\hat{\mathbf{a}}}_{1o} - \ddot{D}_1\hat{\mathbf{a}}_{1o}}{D_1} && (65)\\
\ddot{\hat{\mathbf{a}}}_{3o} &= \frac{\ddot{\mathbf{r}}_{t/so}^{\times}\mathbf{S}_o - 2\dot{D}_2\dot{\hat{\mathbf{a}}}_{3o} - \ddot{D}_2\hat{\mathbf{a}}_{3o}}{D_2} && (66)\\
\ddot{D}_1 &= \frac{(\ddot{\mathbf{r}}_{t/so}\cdot\mathbf{r}_{t/so}) + (\dot{\mathbf{r}}_{t/so}\cdot\dot{\mathbf{r}}_{t/so}) - \dot{D}_1^2}{D_1} && (67)\\
\ddot{D}_2 &= \frac{(\ddot{\mathbf{r}}_{t/so}^{\times}\hat{\mathbf{s}}_o)\cdot(\mathbf{r}_{t/so}^{\times}\hat{\mathbf{s}}_o) + (\dot{\mathbf{r}}_{t/so}^{\times}\hat{\mathbf{s}}_o)\cdot(\mathbf{r}_{t/so}^{\dot{\times}}\hat{\mathbf{s}}_o) - \dot{D}_2^2}{D_2} && (68)
\end{aligned}$$

and then $\dot{\boldsymbol{\omega}}_a^{ai}$ becomes

$$\dot{\boldsymbol{\omega}}_a^{ai} = \mathring{\boldsymbol{\omega}}_a^{ao} + \mathbf{R}^{ao}\dot{\boldsymbol{\omega}}_o^{oi} \tag{69}$$

Likewise, $\mathring{\boldsymbol{\omega}}_a^{ca}$ is found by differentiating the expression in Eq. (51) where

$$\ddot{\theta}_{ac} = \frac{-\ddot{s}_{1a}p_{1c} + 2\ddot{\theta}_{ac}\dot{s}_{2a}p_{2c}\sin\theta_{ac} - \ddot{s}_{2a}p_{2c}\cos\theta_{ac}}{s_{2a}p_{2c}\sin\theta_{ac}} \tag{70}$$

and the acceleration of the sun vector with respect to $\mathcal{F}_a$ is given by

$$\overset{\circ\circ}{\hat{\mathbf{s}}}_a = -2\boldsymbol{\omega}_a^{ai\times}\,\overset{\circ}{\hat{\mathbf{s}}}_a - \boldsymbol{\omega}_a^{ai\times}(\boldsymbol{\omega}_a^{ai\times}\hat{\mathbf{s}}_a) \tag{71}$$

The desired angular acceleration becomes

$$\dot{\boldsymbol{\omega}}_r^{ri} = \mathbf{R}^{ra}(\overset{\circ}{\boldsymbol{\omega}}_a^{ca} + \dot{\boldsymbol{\omega}}_a^{ai}) \tag{72}$$

where as before, $\mathbf{R}^{ra}$ is the rotation matrix from $\mathcal{F}_a$ to $\mathcal{F}_r$. Like the desired angular velocity vector $\boldsymbol{\omega}_r^{ri}$, the desired acceleration vector $\dot{\boldsymbol{\omega}}_r^{ri}$ is constructed from knowing $\overset{\circ}{\boldsymbol{\omega}}_a^{ca}$ and $\dot{\boldsymbol{\omega}}_a^{ai}$ in $\mathcal{F}_r$ using Eq. (55). Having now found $\mathbf{R}^{ri}$, $\boldsymbol{\omega}_r^{ri}$, and $\dot{\boldsymbol{\omega}}_r^{ri}$, we can completely describe the desired trajectory that the real spacecraft needs to have in order to track a target. In the next section, we show how this open-loop reference trajectory is used to derive a control law that will asymptotically drive any initial attitude and rate errors in the body frame to zero.

CONTROLLERS

The nonlinear feedback controller presented in this paper uses momentum wheels to generate the internal axial torque $\mathbf{g}_a$. Here, the wheels are the only devices used to track rigid spacecraft attitude motions and to correct tracking errors. Thrusters were used in Ref. 7 to track the attitude motions while momentum wheels corrected for tracking errors. The only external torque is the gravity gradient torque $\mathbf{g}_e$. Like the controllers used in Ref. 7, this feedback controller globally asymptotically stabilizes the tracking error through the use of a Lyapunov function.

Error Kinematics

We define the tracking error kinematics between the body and reference frames. The attitude tracking error is defined by

$$\mathbf{R}^{br}(\boldsymbol{\delta\sigma}) = \mathbf{R}^{bi}(\boldsymbol{\sigma}_b)\mathbf{R}^{ir}(\boldsymbol{\sigma}_r) \tag{73}$$

with $\mathbf{R}^{br}(\boldsymbol{\delta\sigma})$ being the rotation matrix from the reference frame $\mathcal{F}_r$ to the body frame $\mathcal{F}_b$, and $\boldsymbol{\delta\sigma}$ is the error in the attitude between the frames $\mathcal{F}_b$ and $\mathcal{F}_r$. The tracking error of the angular velocity expressed in $\mathcal{F}_b$ as

$$\boldsymbol{\delta\omega} = \boldsymbol{\omega}_b - \mathbf{R}^{br}(\boldsymbol{\delta\sigma})\boldsymbol{\omega}_r \tag{74}$$

Using Eq. (3), the differential equation for the error kinematics becomes

$$\boldsymbol{\delta\dot{\sigma}} = \mathbf{G}(\boldsymbol{\delta\sigma})\boldsymbol{\delta\omega} \tag{75}$$

With these three relations, we are now ready to derive the control law in the next section.

The Feedback Tracking Controller

The feedback momentum wheel controller is derived from Lyapunov control theory. We use the following Lyapunov function candidate[7]

$$V = \frac{1}{2}\boldsymbol{\delta\omega}^T\mathbf{K}\boldsymbol{\delta\omega} + 2k_2\ln\left(1 + \boldsymbol{\delta\sigma}^T\boldsymbol{\delta\sigma}\right) \tag{76}$$

where k_2 is some positive gain constant. Substituting Eq. (74) into Eq. (76) and taking the derivative, yields the following equation for $\dot{V}$ in terms of $\boldsymbol{\omega}_b$, $\boldsymbol{\omega}_r$, and the tracking errors $\boldsymbol{\delta\sigma}$ and $\boldsymbol{\delta\omega}$:

$$\dot{V} = \left[\dot{\boldsymbol{\omega}}_b - \dot{\mathbf{R}}^{br}(\boldsymbol{\delta\sigma})\boldsymbol{\omega}_r - \mathbf{R}^{br}(\boldsymbol{\delta\sigma})\dot{\boldsymbol{\omega}}_r\right]^T\mathbf{K}\boldsymbol{\delta\omega} + k_2\frac{\boldsymbol{\delta\sigma}^T\boldsymbol{\delta\dot{\sigma}}}{1 + \boldsymbol{\delta\sigma}^T\boldsymbol{\delta\sigma}} \tag{77}$$

We rewrite Eq. (77), using Eqs. (1) and (75), as

$$\dot{V} = \left[\mathbf{J}^{-1}\left(\mathbf{h}_b^\times\boldsymbol{\omega}_b + \mathbf{g}_e - \mathbf{A}\mathbf{g}_a\right) - \boldsymbol{\omega}_b^\times\boldsymbol{\delta\omega} - \mathbf{R}^{br}(\boldsymbol{\delta\sigma})\mathbf{J}^{-1}\left(\mathbf{h}_r^\times\boldsymbol{\omega}_r + \mathbf{g}_e - \mathbf{A}\mathbf{g}_{ar}\right)\right]^T\mathbf{K}\boldsymbol{\delta\omega} + k_2\boldsymbol{\delta\sigma}^T\boldsymbol{\delta\omega} \tag{78}$$

where we have made use of the fact proved in Ref. 7

$$\frac{d\mathbf{R}^{br}(\boldsymbol{\delta\sigma})}{dt}\boldsymbol{\omega}_r = \boldsymbol{\omega}_b^\times\boldsymbol{\delta\omega} \tag{79}$$

Letting $\mathbf{K} = \mathbf{J}$ in Eq. (78) and making use of Eq. (7), the final equation for the derivative of V becomes:

$$\begin{aligned}\dot{V} \;=\; & -\big[-\mathbf{h}_b^\times \mathbf{J}^{-1}(\mathbf{h}_b - \mathbf{A}\mathbf{h}_a) - \mathbf{g}_e + \mathbf{A}\mathbf{g}_a + \mathbf{J}\boldsymbol{\omega}_b^\times \delta\boldsymbol{\omega} \\ & + \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{h}_r^\times \mathbf{J}^{-1}(\mathbf{h}_r - \mathbf{A}\mathbf{h}_{ar}) + \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{g}_e \\ & - \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{A}\mathbf{g}_{ar} - k_2\delta\boldsymbol{\sigma}\big]^T \delta\boldsymbol{\omega} \end{aligned} \tag{80}$$

We want to choose the control torques $\mathbf{A}\mathbf{g}_a$ so that $\dot{V}$ is negative definite. Choosing

$$\begin{aligned}\mathbf{A}\mathbf{g}_a \;=\; & \mathbf{h}_b^\times \mathbf{J}^{-1}(\mathbf{h}_b - \mathbf{A}\mathbf{h}_a) + \mathbf{g}_e - \mathbf{J}\boldsymbol{\omega}_b^\times \delta\boldsymbol{\omega} \\ & - \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{h}_r^\times \mathbf{J}^{-1}(\mathbf{h}_r - \mathbf{A}\mathbf{h}_{ar}) - \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{g}_e \\ & + \mathbf{J}\mathbf{R}^{br}(\delta\boldsymbol{\sigma})\mathbf{J}^{-1}\mathbf{A}\mathbf{g}_{ar} + k_1\delta\boldsymbol{\omega} + k_2\delta\boldsymbol{\sigma} \end{aligned} \tag{81}$$

where k_1 is a positive gain constant leads to

$$\dot{V} = -k_1 \delta\boldsymbol{\omega}^T \delta\boldsymbol{\omega} \leq 0 \tag{82}$$

As found in Ref. 7, it can be shown that the control law in Eq. (81) guarantees perfect tracking, i.e., $\boldsymbol{\omega}_b(t) = \boldsymbol{\omega}_r(t)$ and $\boldsymbol{\sigma}_b(t) = \boldsymbol{\sigma}_r(t)$ for all $t \geq 0$ if the initial condition errors are zero, i.e., $\delta\boldsymbol{\omega}(0) = \delta\boldsymbol{\sigma}(0) = 0$.

NUMERICAL EXAMPLE

We demonstrate the capability of the momentum wheel control law by presenting a target tracking example. Given a circular orbit with an altitude of 279.24 km, we want to a spacecraft to acquire and track Cape Canaveral, Florida ($\delta_t = 28.467°N$, $L_t = 80.467°W$) starting with a sub-satellite point located at $19.583°N$ longitude and $118.381°W$ longitude. Here, our algorithms do not determine whether or not the target is actually visible by the sensor. We will assume, for this example, that the spacecraft can see the target. The spacecraft is modeled as a gyrostat with three momentum wheels aligned with the principal axes. Their axial moments of inertia are given as

$$\mathbf{I}_s = \mathrm{diag}(10, 30, 70) \tag{83}$$

We let the spacecraft total moment of inertia matrix (platform and momentum wheels) be

$$\mathbf{J} = \mathrm{diag}(200, 150, 175) \tag{84}$$

and the solar panel unit vector $\hat{\mathbf{p}}$ is defined in $\mathcal{F}_b$ as

$$\hat{\mathbf{p}} = [0.5437 \;\; 0.8269 \;\; 0.1440]^T \tag{85}$$

The spacecraft actual initial attitude is $\boldsymbol{\sigma}_b(0) = [-0.1259 \;\; 0.2598 \;\; -0.0988]^T$ with its sensor boresight initially pointing at $24.737°N$ longitude and $100.435°W$ longitude. We let the platform initially rotate with $\boldsymbol{\omega}_b(0) = [-0.0040 \;\; -0.00854 \;\; 0.0009]^T$.

The target tracking maneuver results are shown in Figs. 3, 4, and 5. The gains[7] for the controller were chosen to be $k_1 = 54$ and $k_2 = 47$.

Figure 3(a) shows the time history of $\delta\boldsymbol{\sigma}$, and Figure 3(b) shows the time history of $\delta\boldsymbol{\omega}$. It can be seen that the controller does indeed make the body frame track the prescribed reference trajectory. All of the attitude and rate errors were driven asymptotically to zero over time. Figure 4(a) shows the control torque generated by the controller. The controller initially generates large torques to slew the spacecraft to point at the target. Once the target has been acquired, which means all errors are zero, the torques needed to track the target become very small and equal to $\mathbf{g}_{ar}$. Figure 4(b) illustrates the ideal torque $\mathbf{g}_{ar}$ needed for target tracking if all there are no initial errors. Figure 5 illustrates the yaw-steering condition of Eq. (16). It is clearly seen that the solar panel axis becomes normal to the sun vector about the same time the spacecraft acquires the target (40 sec.). Thus, Figure 5 indicates that the spacecraft can simultaneously a target while maintaining its power requirements.

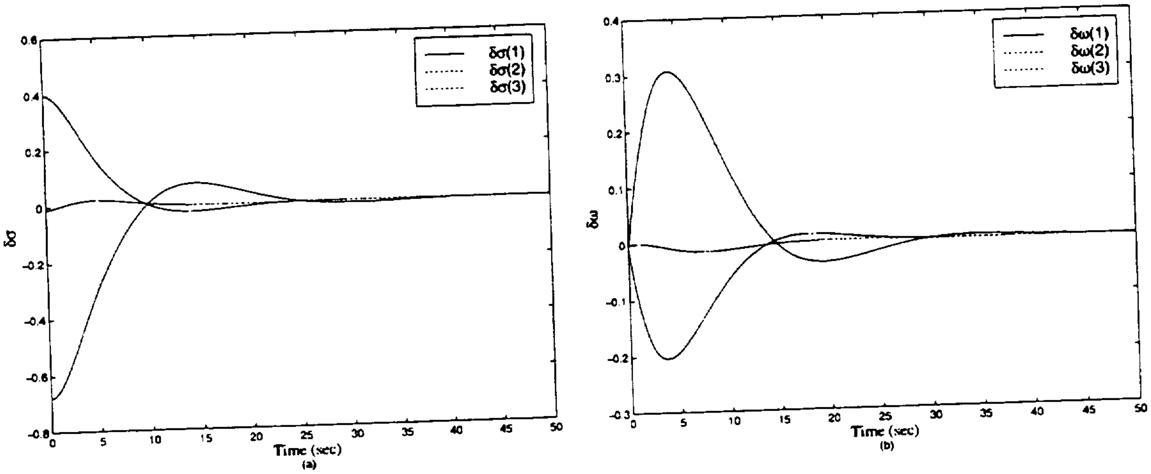


Figure 3: The time history of (a) $\delta\boldsymbol{\sigma}$ and (b) $\delta\boldsymbol{\omega}$.

CONCLUSIONS

A method for computing a multiaxis target tracking trajectory has been developed that also allows the solar panel normal to be aligned with the sun vector during a tracking maneuver. Other authors have developed similar algorithms, but the open-loop trajectory in this paper is attitude parameter independent with the sun tracking requirement incorporated. A control law has been developed that uses internal torques provided by the momentum wheels for tracking rotational maneuvers. The control law is a function of the spacecraft angular momentum, wheel momenta, and attitude, as well as the desired pointing direction. A simple tracking maneuver example clearly shows that the wheel controller makes the body frame track the reference motion.

REFERENCES

[1] Gramling, C. J., Lee, T., Niklewski, D. J., and Long, A. C., "Relative Navigation For Autonomous Formation Flying Of Spacecraft," In *Proceedings of the AAS/AIAA 1997 Astrodynamics Specialist Conference*, 1997.

[2] DeCou, A. B., "Orbital Station-Keeping for Multiple Spacecraft Interferometry," *Journal of the Astronautical Sciences*, Vol. 39, No. 3, 1991, pp. 283–297.

[3] Ulybyshev, Y., "Long-Term Formation Keeping of Satellite Constellation Using Linear-Quadratic Controller," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 109–115.

[4] Folta, D., Bordi, F., and Scolese, C., "Considerations On Formation Flying Separations For Earth Observing Satellite Missions," *Advances in Astronautical Sciences*, Vol. 79, No. 2, 1992, pp. 803–822.

[5] Hablani, H. B., "Design of a Payload Pointing Control System for Tracking Moving Objects," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 3, 1989, pp. 365–374.

[6] Hablani, H. B., "Multiaxis Tracking and Attitude Control of Flexible Spacecraft with Reaction Jets," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, 1994, pp. 831–839.

[7] Hall, C. D., Tsiotras, P., and Shen, H., "Tracking Rigid Body Motion Using Thrusters and Momentum Wheels," In *1998 AIAA/AAS Astrodynamics Conference*, August 1998.

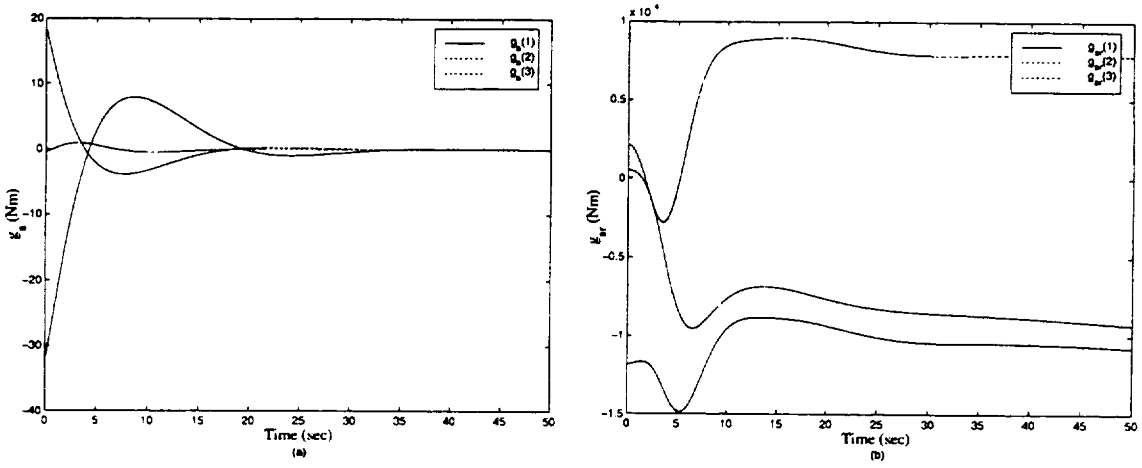


Figure 4: (a) The momentum wheel feedback control law in Eq. 81, and (b) The desired control for target tracking.

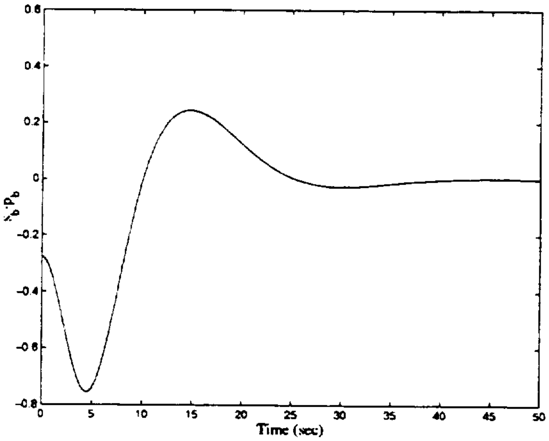


Figure 5: The yaw-steering condition of Eq. (16).

[8] Schaub, H., Robinett, R. D., and Junkins, J. L., "Globally Stable Feedback Laws for Near-Minimum-Fuel and Near-Minimum-Time Pointing Maneuvers for a Landmark-Tracking Spacecraft," *Journal of the Astronautical Sciences*, Vol. 44, No. 4, 1996, pp. 443–466.

[9] Steyn, W. H., "Near-Minimum-Time Eigenaxis Rotation Manuevers Using Reaction Wheels," *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 5, 1995, pp. 1184–1189.

[10] Wie, B. and Lu, J., "Feedback Control Logic for Spacecraft Eigenaxis Rotations Under Slew Rate and Control Contraints," *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 6, 1995, pp. 1372–1379.

[11] Hughes, P. C., *Spacecraft Attitude Dynamics*, John Wiley & Sons, New York, 1986.

[12] Hall, C. D., "Spinup Dynamics of Gyrostats," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 1177–1183.

[13] Wertz, J., editor, *Spacecraft Attitude Determination and Control*, D. Reidel, Dordrecht, Holland, 1978.

[14] Shuster, M. D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 439–517.

[15] Kalweit, C. C., "Optimum Yaw Motion for Satellites with a Nadir-Pointing Payload," *Journal of Guidance, Control, and Dynamics*, Vol. 6, No. 1, 1983, pp. 47–52.

[16] Meirovitch, L., *Methods of Analytical Dynamics*, McGraw-Hill, New York, 1970.

[17] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, New York, 1997.

[18] Meriam, J. L. and Kraige, L., *Dynamics (SI Version)*, John Wiley & Sons, New York, third edition, 1992.

DEVELOPMENT OF FORMATION DEPLOYMENT AND INITIALIZATION CONCEPTS

By

Surjit S. Badesha
Surjit.Badesha@jhuapl.edu
Gene A. Heyler
Peter J. Sharer
Thomas E. Strikwerda

Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel MD 20723-6099

ABSTRACT

NASA's Cross-Cutting Technology Development Program identified formation flying as a key enabler for the next generation Earth and Sciences campaign. It is hoped that this technology will allow a distributed network of autonomous satellites to act collaboratively as a single collective unit paving the way for extensive co-observing campaigns, coordinated multi-point observing programs, improved space-based interferometry, and entirely new approaches to conducting science. APL as a team member with GSFC, funded by the Earth Sciences and Technology Organization (ESTO), investigated formation deployment and initialization concepts which is central to the formation flying concept. This paper presents the analytical approach and preliminary results of the study.

The study investigated a simple mission involving the deployment of six micro-satellites, one at a time, from a bus. At the initialization state, the satellites fly in an along-track trajectory separated by nominal spacing. The study entailed the development of a two-body (bus and satellite) relative motion propagator based on Clohessy-Wiltshire (C-W) equations with drag from which the relative motion of the micro-satellites is deduced. This code was used to investigate cluster development characteristics subject to "tip-off" (ejection) conditions. Results indicate that cluster development is very sensitive to the ballistic coefficients of the bus and satellites, and to relative ejection velocity. This information can be used to identify optimum deployment parameters, along with accuracy bounds for a particular mission, and to develop a cluster control strategy minimizing global fuel and cost. A suitable control strategy concept has been identified, however, it needs to be developed further.

INTRODUCTION

To advance space-based scientific research within NASA's budgeting constraints, numerous distributed multi-vehicle concepts for future space missions are being proposed. It is hoped that the multi-vehicle approach will allow a distributed network of autonomous vehicles to act collaboratively as a single collective unit paving the way for extensive co-observing campaigns, coordinated multi-point observing programs, improved space-based interferometry, and entirely new approaches to conducting science.

Distributed, multi-vehicle concepts fall into two categories; constellation and formation flying [1]. A constellation is defined as two or more spacecraft in similar orbits that have separate control of their orbits. Accurate knowledge of the spacecraft state is generally required for post-processing of multi-point observational data. However, there is no hard requirement to maintain relative position or velocity between the spacecraft. The Auroral Multi-scale Midex (AMM), and the Multi-point Magnetospheric Reconnaissance Imaging proposals for spatially distributed samplings of Earth's Aurora and Magnetosphere, respectively, are good examples of constellation concepts. The Globalstar and Iridium systems for global telephony are other examples from the commercial arena.

Formation flying, on the other hand, involves maintaining a distance between one or more vehicles in addition to following a referenced trajectory. Maintenance of relative separation within a predetermined box requires active control of the spacecraft. This requires real-time interspacecraft communication of spacecraft state and attitude so that a cluster of spacecraft can operate autonomously. Interspacecraft communication is critical to formation flying. Stadter at APL [2], and How at Stanford University [3], are actively researching interspacecraft communication based on a differential GPS system. They reported some success based on their preliminary investigation. The AFRL TechSat 21 program that envisions the use of sparse aperture radar for ground moving target indication, Das [4], and ORION, a low-cost demonstration of formation flying in space using GPS, are good examples of the formation flying concept.

Both constellation and formation flying require deployment and initialization before initiating the scientific operation phase of the mission. There are a number of deployment strategies that have been proposed and used in the past, such as ejecting one satellite at a time axially from a dispenser (Iridium, TIMED, and AMM missions) and radial separation of the satellites away from a central core dispenser (Globalstar). A deployment strategy is highly dependent on the launch vehicle and the initialization condition requirements of the mission. In general, the primary design driver for any satellite system optimization is to minimize the fuel required during the initialization and operational phases of the mission, since additional fuel translates into increased payload weight and size, resulting in higher cost. According to the AMM proposal, about two-thirds of the total fuel is required for the initialization phase of the mission. Therefore, optimization of both the deployment and cluster control strategies for initialization is thought to be central to the mission.

Per GSFC direction, on behalf of the ESTO, we investigated a modified Orion mission requiring deployment of six micro-satellites, one at a time, from a bus. At the initialization state the satellites fly in an along-track trajectory separated by nominal spacing. The objective of the study was to investigate cluster development characteristics subject to "tip-off" conditions. This information can be used to identify optimum deployment parameters, along with accuracy bounds for a particular mission, and to develop a cluster control strategy minimizing fuel and cost.

TECHNICAL APPROACH

Constellation/formation flying allows multiple-point viewing for advanced scientific space missions. The goals of each mission can vary widely. Therefore, it is necessary to clearly define the scientific mission. This would lead to a preliminary definition of the satellite system (i.e. instruments, number of satellites, orbital characteristics).

There are three primary phases associated with constellation/formation flying, namely deployment, initialization, and scientific operation. The major design objective is to accomplish the scientific mission at minimal cost (see Figure 1). Therefore, an optimal satellite system needs to be as lightweight as possible to keep launch costs down. One way to minimize weight substantially is to minimize the fuel requirements for the initialization and scientific operational phases of the mission. Therefore, fuel efficiency is a major design driver.

In recent years, the operational phase of various missions has received considerable attention [1, 3, 4, 5, 6, 7, 8]. However, despite the obvious importance of the deployment and initialization phases, these phases have been overlooked. Therefore, the goal of this research is to develop optimal deployment and control strategies for initialization for various potential missions. Optimization is dependent on a number of variables, including satellite capabilities and "tip-off" conditions.

The approach taken to optimize deployment and control strategies for initialization is presented in Figure 2. As shown, it is necessary to first gain an understanding of cluster development characteristics subject to "tip-off" conditions (Step 1). This entailed developing/identifying algorithms and simulation tools for the analytical studies. These tools are utilized to investigate a modified simple Orion mission. Upon gaining an understanding of relevant issues and the development of an analytical approach, more complex missions requiring more complicated deployment strategies, as well as higher fidelity simulations, will be considered (Step 2). This step should result in the formulation of a matrix for specifying formation flying

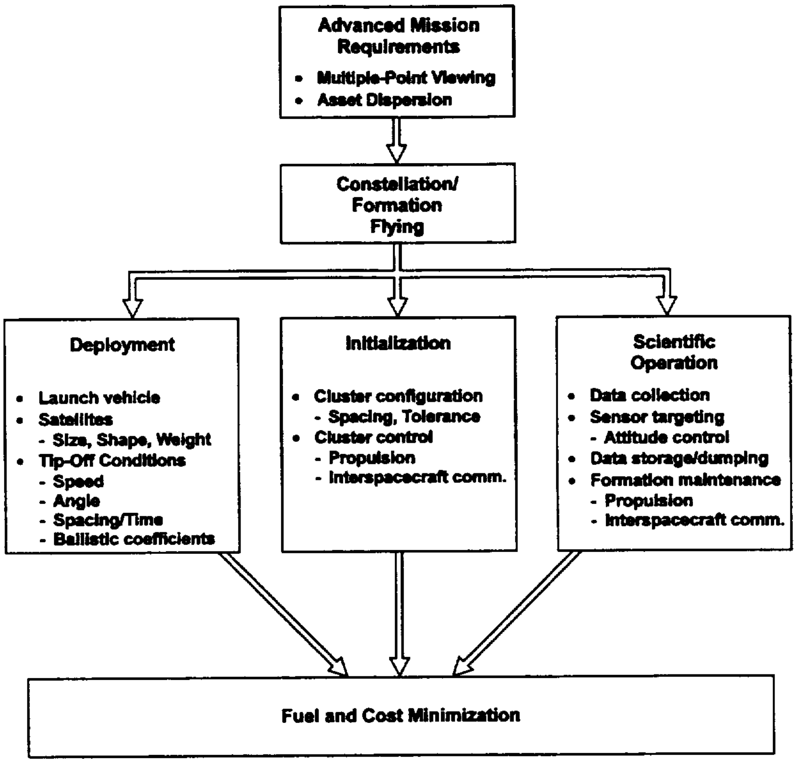


Figure 1. Formation Flying Functional Flow Chart

STEP 1 – Gain understanding and develop an approach.

- Start with simplest modified Orion mission.
  - Deploy six micro-satellites from a bus, one at a time
  - Initialization state has the satellites flying in an along-track trajectory separated by 100 m within 5 m tolerance.
- Develop two-body relative motion propagator based on Clohessy-Wiltshire equations and deduce relative motion of satellites.
  - Include drag perturbation
  - Exclude earth's oblateness, and solar radiation force effects
- Evaluate propagator's performance.
- Investigate deployment and cluster development characteristics subject to "tip-off" conditions.

STEP 2 – Consider complex missions.

- Investigate more complex missions, deployment configurations, control strategies, and higher fidelity simulations.
  - Non-coplanar
  - Secular orbital drift
  - Elliptical orbit
  - Radial deployment
  - Stack deployment
  - High fidelity propagator
- For 2-D and 3-D configurations, formulate metric for specifying formation.

Simplistic:	Intra-clustering spacing at epoch
Better:	Spacing plus derivatives (expanding/collapsing) at epoch
Best:	Spacing and derivatives at epoch plus long term secular growth

STEP 3 – Optimization of cluster control strategies.

- Develop cluster control strategy.
  - Use pattern matching technique in a relative distance matrix (global fuel minimization)
  - Two impulse burn scheme needed for "better" and "best" approach
  - Necessitates long term orbit propagator with J2 zonal gravity model, eccentricity, and drag states
  - Evaluate ballistic coefficient control

Figure 2. Approach to the Development of Deployment and Control for Initialization Strategies for Formation Flying Missions

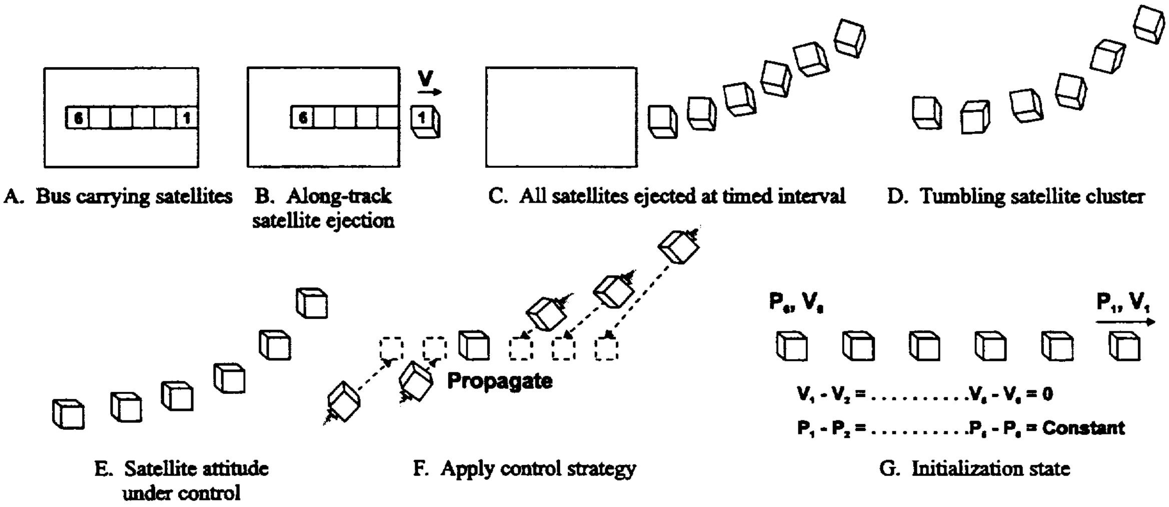
requirements for various missions. The final task (Step 3) is to develop optimal cluster control strategies for various missions. In some cases a control strategy for a given mission can be used for both the initialization and formation maintenance phases of the mission.

The steps outlined in Figure 2 should result in a good understanding of the relevant issues, leading to the identification of a suitable deployment and control strategy for a given mission. Due to funding limitations, research associated with only the first step has been accomplished. A deployment concept, control strategy, simulation tools, and the preliminary results are presented in the next four sections.

DEPLOYMENT CONCEPT

The original Orion mission has a stack of three or six satellites being ejected from a dispenser in a co-planar, circular orbit [3]. It is assumed that the stack(s) could be ejected axially or radially, depending on the size of the micro-satellites and launch vehicle, using currently available technology. Thereafter, the satellites will be separated in steps while being under control to finally end up with the initialization state. The initialization state has the six satellites in a cluster, flying in an along-track trajectory separated by 100 m with 5 m tolerance.

Knowing that a Delta II bus is capable of ejecting one satellite at a time with predefined "tip-off" conditions, we developed a modified deployment concept and control strategy for initialization, as illustrated in Figure 3. This concept has a bus carrying six micro-satellites, being ejected one at a time at timed intervals at a predefined relative velocity from the bus. All the satellites travel along the same trajectory with respect to the bus. The tumbling satellites in a cluster are expected to be under attitude control within six hours to a number of days, depending on the active or passive momentum dump capability aboard the satellites. Thereafter, a control strategy is applied to propagate the satellites to the required initialization state, utilizing the appropriate amount of thrust. This concept utilizes the full capabilities of the bus, thereby simplifying the satellite's stack separation mechanism requirement. This approach would contribute to lowering overall deployment cost.


6
1
6
V
1
Propagate
P6, V6
P1, V1
V1 - V2 = ...V5 - V6 = 0
P1 - P2 = ...P5 - P6 = Constant

A. Bus carrying satellites B. Along-track satellite ejection C. All satellites ejected at timed interval D. Tumbling satellite cluster

E. Satellite attitude under control F. Apply control strategy G. Initialization state

Figure 3. Deployment Concept and Control Strategy

CONTROL STRATEGY

Strategy for controlling the cluster can be separated into two independent tasks - (1) the intra-cluster spacing that defines the geometry, and (2) the orbital position of the overall cluster itself. The latter is mission dependent and may or may not be a requirement; it becomes necessary only if there are time dependent requirements such as maintaining a repeat ground track or synchronizing with other spacecraft outside the cluster.

One favorable approach for controlling the intra-cluster spacing is to apply simple pattern matching techniques to a relative distance matrix. For example, a six member cluster would be characterized by a 6x6 matrix of distances with a necessarily zero diagonal (since each spacecraft is by definition zero distance from itself). By inspecting the rows and columns of the matrix, a global solution can be determined specifying how much each satellite must be maneuvered. This has advantages over the "mother ship" approach in which each satellite is maintained relative to a particular one designated as the "mother ship". For example, if only the mother ship was out of position, then the latter method would force all other satellites to maneuver relative to it; however, the matrix method would isolate the mother ship as being out of position and only it would maneuver. This method will minimize overall cluster fuel usage. Figure 4 is a conceptualization that contrasts the two approaches.

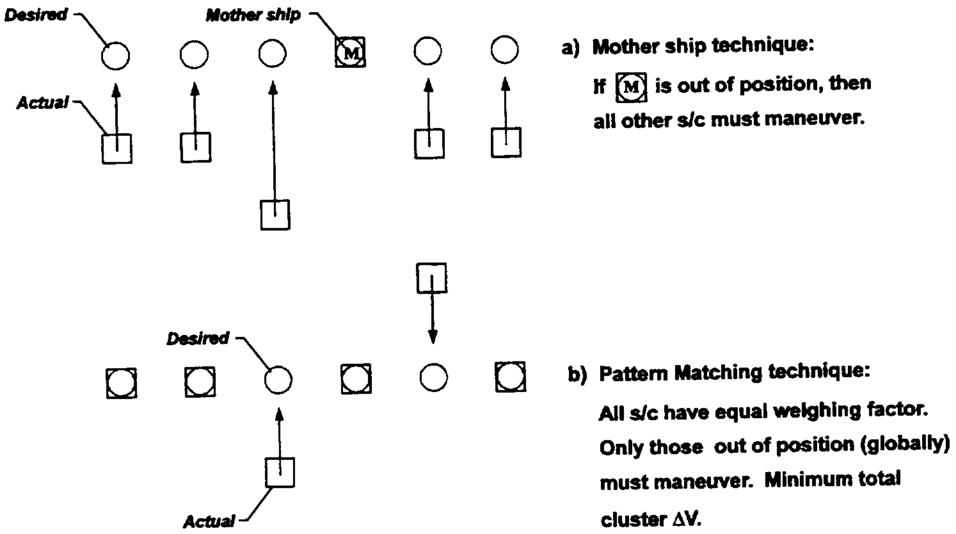

Desired
Mother ship
M
Actual
a) Mother ship technique:
If M is out of position, then
all other s/c must maneuver.
Desired
b) Pattern Matching technique:
All s/c have equal weighing factor.
Only those out of position (globally)
must maneuver. Minimum total
cluster ΔV.
Actual

Figure 4. Conceptualization of the Two Control Techniques (Mother Ship vs. Pattern Matching)

Cluster geometry could be controlled by uplinking a small vector or matrix to each of the identical satellites. For example, the 1-D case might be defined simply by:

[0, d2, d3, d4, d5, d6, T]

The first six numbers represent the desired along-track distances of each satellite relative to the first. The parameter T is a time dependent variable (such as UT time, true anomaly, argument of latitude, etc.) that specifies when the distances are to be defined. For simple circular orbits, T would not be a requirement, but it would be for eccentric orbits.

We retain the leading slot (currently zero) in the vector as a potential mechanism for specifying relative movement of the entire cluster forward or backward in orbit, thus possibly combining the two independent control tasks into one unified algorithm.

The 3-D case might be controlled by a matrix, such as:

[dh1, dl1, dc1, T1]
[dh2, dl2, dc2, T2]
[0, 0, 0, 0]
[dh4, dl4, dc4, T4]
[dh5, dl5, dc5, T5]
[dh6, dl6, dc6, T6]

In this example, each satellite would be specified at some H-L-C position relative to the third at some point on the orbit. This concept might be extended to include a relative velocity state as well. We believe an extended and enhanced form of C-W equations [4] that include eccentricity, drag, and J2 are needed for the control decisions. This is driven, in part, by the instantaneous GPS data which will produce a non-constant orbit period calculation, and by any relative cross-track components which imply a possible orbit inclination difference which leads to secular cross-track growth. The simple form of C-W equations generally will not be adequate for predicting future relative positions in the cluster control algorithms.

SIMULATION TOOL

For the modified Orion mission analysis, a C-W equation based relative motion propagator for a circular orbit was developed in the MATLAB environment.

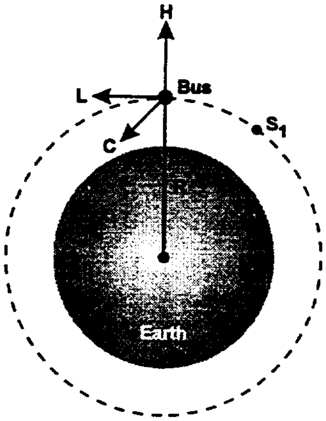


Figure 5. Relative Motion Coordinate System

Referring to the relative motion coordinate system shown in Figure 5, if the reference vehicle (Bus) and satellite (S1) orbits are both nearly circular, then relative motion can be expressed as:

$$\ddot{H} - 2n\dot{L} - 3n^2H = f_H \tag{1}$$

$$\ddot{L} + 2n\dot{H} = f_L \tag{2}$$

$$\ddot{C} + n^2C = f_C \tag{3}$$

and $$n = \sqrt{\mu / R^3} \tag{4}$$

where μ = GM of the Earth
R = Reference radius

and f_H, f_L, f_C = Acceleration due to perturbation along three axes.

By assuming a nearly circular orbit, the earth's oblateness effect is excluded. Furthermore, by limiting the reference altitude to less than 500 km, it is reasonable to exclude the solar radiation force since it is an order of magnitude smaller than the drag force per unit area at this altitude. Including only the drag perturbation force, which is always in the negative L direction, then:

$$f_H = f_C = 0 \tag{5}$$

and $$f_L = -1/2\, \rho\, V^2/B_R \tag{6}$$

where ρ is the density at the reference attitude, assumed constant for the duration of deployment and the initialization phase of the flight, V is the along-track velocity and B_R is the relative ballistic coefficient defined as:

$$B_R = \frac{B_B \cdot B_S}{B_R - B_S} \tag{7}$$

where subscript B and S stand for the reference bus and a satellite, respectively.

The second order differential equations (1) through (3) along with the drag force acceleration, represented by equation (6), have the following analytical solutions in matrix form:

$$\begin{bmatrix} H \\ L \\ C \\ \dot{H} \\ \dot{L} \\ \dot{C} \end{bmatrix} = \begin{bmatrix} (4-3C) & 0 & 0 & S/n & 2(1\text{-}C)/n & 0 & k/(S/n - t)n \\ 6(S - nt) & 1 & 0 & 2(C\text{-}1)/n & (4S/n - 3t) & 0 & k(2C/n^2 + 3t^2/4 - 2/n^2) \\ 0 & 0 & C & 0 & 0 & S/n & 0 \\ 3nS & 0 & 0 & C & 2S & 0 & k/(C-1)n \\ 6n(C-1) & 0 & 0 & -2S & (4C-3) & 0 & k(-2S/n + 3t/2) \\ 0 & 0 & -nS & 0 & 0 & C & 0 \end{bmatrix} \begin{bmatrix} H_0 \\ L_0 \\ C_0 \\ \dot{H}_0 \\ \dot{L}_0 \\ \dot{C}_0 \\ 1 \end{bmatrix} \tag{8}$$

where C = Cos(nt), S = Sin(nt), $k = \rho\mu/RB_R$, t = time, and subscript 0 implies initial condition "tip-off" values.

The explicit analytical solution, expressed by equation (8) is the basis of our propagator, programmed in the MATLAB environment. Given the ballistic coefficients of two bodies and the initial state vector, the propagator predicts relative motion at a future time. The relative motion trajectory is generated by making calculations at a number of time steps. For the modified Orion mission, motion trajectory relative to the bus is generated for the six satellites. Knowing the separation time between the satellites, the state of each satellite with respect to any other can be determined. This methodology is used to investigate the effect of ejection speed, separation time, and the ballistic coefficient on the cluster development.

PRELIMINARY RESULTS

A. Bus-Satellite Relative Trajectory and Pertinent Issues

Consider a scenario where six micro-satellites are ejected from the Delta II bus with an along-track velocity of 0.1 m/s and separated by 100 sec. The satellites may be ejected from the bus by "pushing-off" utilizing a pyro/spring mechanism. Current spring technology is capable of providing "push-off" speed ranging from 0.1 to 5 m/s. After deployment, to keep the satellites in a cluster as tight as possible, minimal possible deployment speed and separation time between satellites are considered. The bus and satellites are assumed to have a mass and frontal area of 950 kg, 40 kg, and 4.52 m^2, 0.375 m^2, respectively. The satellite average cross-sectional area, used in place of the frontal area, represents the tumbling phase of the satellite trajectory.

With respect to the bus, all satellites start off in the forward along-track (+L) and radially outward (+H) direction, then propagate backward and downward, forming a loop at each integer orbit period as shown in Figure 6a. Initially, the size of the loop grows, but after six days it starts to contract. This orbital dynamic phenomena raises two issues pertinent to formation flying:

1. Do the satellites ever collide, and if so, how long after deployment?
2. Is the first loop big enough to avoid satellite collision?

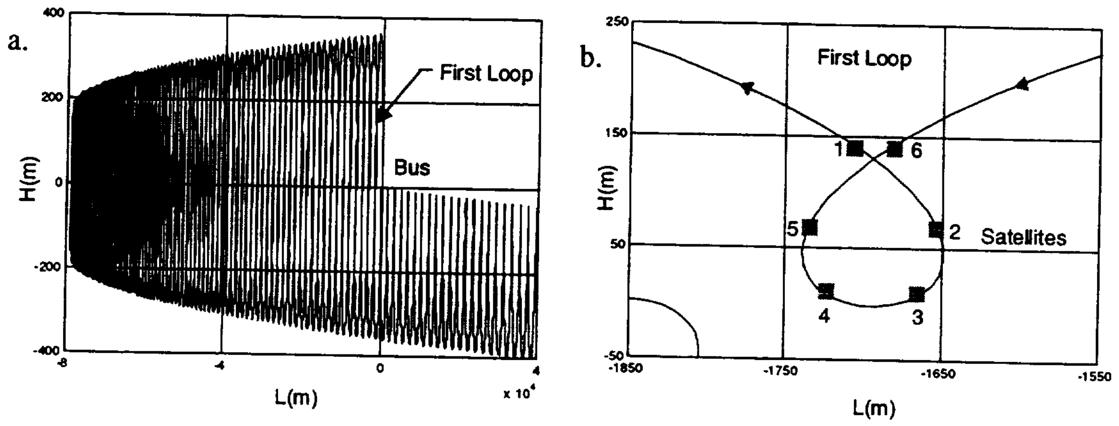


Figure 6. Bus-Satellite Relative Trajectory and Pertinent Issues

Figure 7 shows the initial relative motion between the satellites with respect to satellite 1. Each satellite is in an elliptical orbit in the H – L plane. Due to the drag force effect, spacing between the satellites appears to be decreasing. Spacing between adjacent satellites as a function of time is plotted in Figure 8. Spacing oscillates between 10 m and 70 m with the orbital period. Amplitude of the oscillation decreases with time, but starts to increase after six days. After approximately fourteen days, the satellites can potentially collide as illustrated in Figure 8. Typically, satellite tumbling motion due to asymmetric ejection is brought under control within a day, if not a few hours. Therefore, a cluster control strategy could be implemented long before the satellites collide. Hence, satellite collision is not a major concern for this deployment condition.

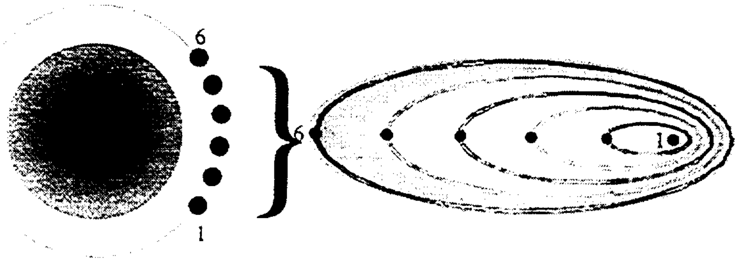


Figure 7. Relative Motion Between Satellites in Formation Flying

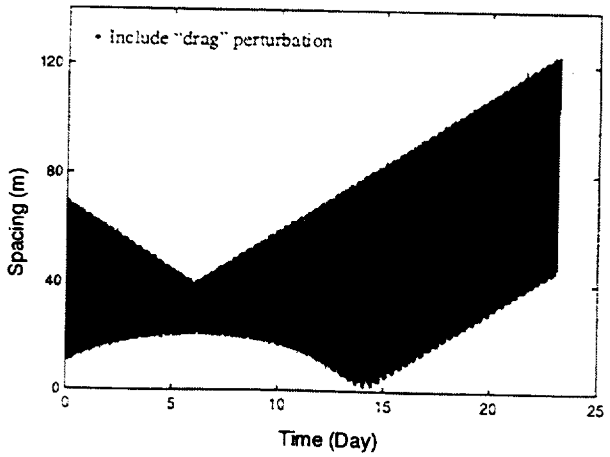


Figure 8. Spacing Between the Adjacent Satellites with the C-W Propagator

Referring to Figure 6b, the first loop length is approximately 325 m. About 8 percent of the time when the satellite spacing is between 65 and 70m, the first loop may not be big enough to avoid a potential collision, since the first satellite in the loop does not cross the trajectory path before the last satellite, as illustrated in Figure 6b. Potential collision can be avoided in one of two ways; by slightly reducing the time between satellite ejection or by increasing ejection speed.

B. Assessment of the C-W Solution Based Propagator

The above analysis is strongly dependent on the accuracy of the orbital simulator. Therefore, the C-W propagator simulation with the associated assumptions is compared with the high precision McDonnell-Douglas BG-14 orbit propagator simulation. The BG-14 propagator takes into consideration solar pressure, oblateness of the earth (up to 40x40) and air density variation with the Jacchia 70 (J70) model [10]. Figure 9 shows the satellite spacing graph produced with the BG-14 simulator for the previous case, corresponding to Figure 8. The two simulations have the same features, however, the BG-14 predicts collision time after about 21 days. The C-W propagator appears to be adequate for the sensitivity studies, as well as for predicting, reasonably accurately, relative motion a short time after ejection. This is good enough for preliminary analysis, however, a higher fidelity propagator should be used for an accurate system analysis.

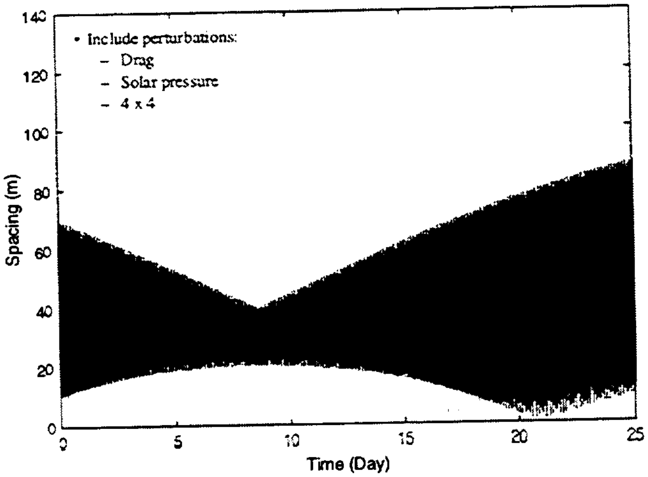


Figure 9. Spacing Between the Adjacent Satellites with the BG-14 Propagator

C. Relative Ballistic Coefficient

The difference in the ballistic coefficients of the bus and satellites results in a relative trajectory as shown in Figure 6a, causing potential collision between ejected satellites. However, if the bus and satellites have the same ballistic coefficient, then the trajectory characteristics would not change with the orbital period as shown in Figure 10. Consequently, the satellites would continue to separate. Theoretically, the difference in the ballistic coefficients could be used to control satellite collision time. However, because of other important design considerations, this would not be done.

D. Realistic "Tip-Off" Conditions

So far we have looked at cluster development subject to the ballistic coefficient. We have not considered effects of the ejection speed and angle, time between spacecraft ejection, and the difference in the ballistic coefficients of the spacecraft. Furthermore, even if optimal values for all the variables are identified for a particular mission there is an issue of accuracy in implementation. For example, the spring mechanism is believed to be able to control the release speed to within two to ten percent one sigma variation, depending on how much effort is put into the calibration on the ground.

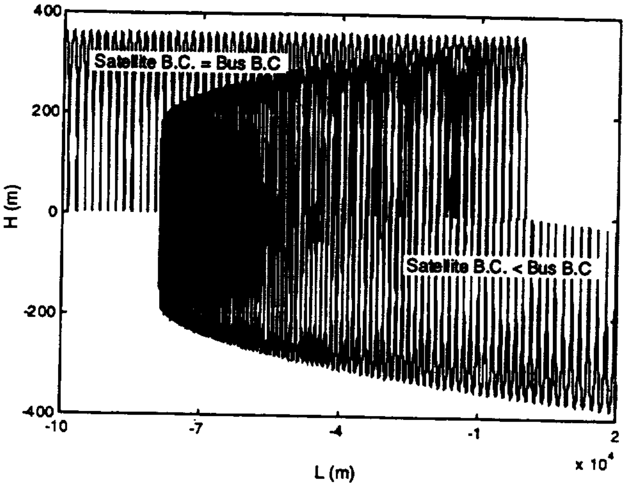


Figure 10. Effects of Relative Ballistic Coefficients of Bus and Satellite

To understand the above issues, consider a scenario where the satellites are ejected by a randomly picked speed ranging from 0.09 to 0.11 m/s, i.e. with the 10 percent one sigma value of 0.1 m/s nominal speed. The reference satellite 1 is ejected with the nominal speed. As shown in Figure 11, two days after deployment the cluster size grows to over 5 km, compared to about 0.3 km if all the satellites were released at the speed of 0.1 m/s. Cluster size is strongly dependent on which of the satellites has greater variation from the nominal speed. Logically, the fuel required to initialize the cluster in this scenario will be significantly higher than if all the satellites were released with the same nominal speed. The perturbation speed prevents satellite collision after fourteen days for the same ejection speed, as noted in Section A. However, closer to deployment time, the perturbation increases the probability of collision, as shown by the inset in Figure 11.

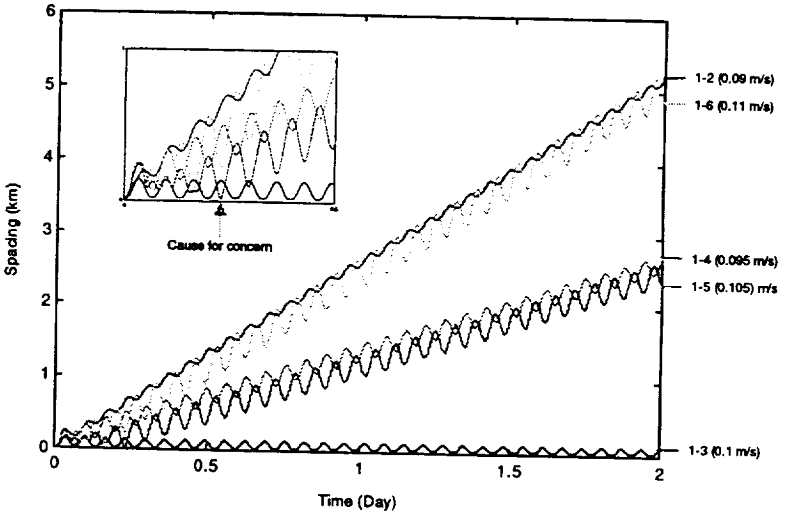


Figure 11. Effects of Realistic Cluster Ejection Speeds

Variation in the satellite ballistic coefficient, launch angle, and separation time would make things worse. A Monte-Carlo type of simulation, with Gaussian distribution for pertinent variables, could be done to determine a realistic fuel requirement for the initialization phase of a particular mission. As implied by the last scenario, there could be a significant difference between the fuel requirement due to inclusion and exclusion of the perturbation consideration. Perturbation needs to be minimized to reduce the fuel requirement during the initialization process. However, in practice this desire needs to be balanced against the engineering effort required for overall cost minimization.

CONCLUSIONS

A first order analytical tool and process has been developed to investigate issues associated with the deployment and initialization phases of a co-planar, circular orbit formation flying mission. Preliminary results indicate that cluster development is very sensitive to the ballistic coefficient of the bus and satellites, and to "tip-off" conditions. The process developed could be used to optimize the deployment and initialization phases of the formation flying mission, from both a fuel and overall cost point of view. A higher fidelity propagator with greater capability, such as the BG-14, should be used for accurate system analysis and to model more complex missions involving non-coplanar and elliptical orbits, and stacked and radial deployment configurations. Finally, because of fuel and cost considerations, both the deployment concept and control strategy for initialization are central to the formation flying mission.

ACKNOWLEDGEMENT

This work was funded by NASA GSFC on behalf of ESTO with David Weidow as the point of contract.

REFERENCES

1. D.C. Folta, L.K. Newman and T. Gardner. "Foundations of Formations Flying for Mission to Planet Earth and New Millennium." AIAA – 96 – 3645 – CP.

2. P.A. Stadter, W.S. Devereux, R.A. Denissen, D.J. Duven, M.S. Asher. "Interspacecraft Communications for Formation Flying," NASA/ESTO Study Report. JHU-APL, January '99.

3. J.P. How, R. Twiggs, D. Weidow, K. Hartman, and F. Bauer. "Orion: A Low Cost Demonstration Formation flying in Space using GPS." AIAA 1998 – 4398.

4. A. Das and R. Cobb. "TechSat 21: Space Missions using Collaborating Constellations of Satellites." 12th AIAA/USU Conference on Small Satellites, Vol. VI-1, 1998.

5. M. Mathews and S.J. Laszkiewicz. "Efficient Spacecraft Formation Keeping With Consideration of Ballistic Coefficient Control." AIAA-88-0375.

6. G.J. Yashko, D.E. Hastings. "Analysis of Thruster Requirements, and Capabilities for Local Satellite Clusters," AIAA, 1996.

7. D. Folta, L.K. Newmann, and D. Quinn. "Design and Implementation of Satellite Formations and Constellations, AAS-98-304.

8. D. Folta, "Enhanced Formation Flying for the New Millenium and Mission to Planet Earth Progress." NASA-GSFC.

9. W.H. Clohessy and R.S. Wiltshire. "Terminal Guidance System For Satellite Rendezvous." JAS Sept. 1960.

10. McDonnell Douglas Aerospace "BG-14 User's Manual." March 1997.

FEASIBILITY OF DECENTRALIZED LINEAR-QUADRATIC-GAUSSIAN CONTROL OF AUTONOMOUS DISTRIBUTED SPACECRAFT

J. Russell Carpenter
NASA Goddard Space Flight Center

ABSTRACT

A distributed satellite formation, modeled as an arbitrary number of fully connected nodes in a network, could be controlled using a decentralized controller framework that distributes operations in parallel over the network. For such problems, a solution that minimizes data transmission requirements, in the context of linear-quadratic-Gaussian (LQG) control theory, was given by Speyer [1]. This approach is advantageous because it is non-hierarchical, detected failures gracefully degrade system performance, fewer local computations are required than for a centralized controller, and it is optimal with respect to the standard LQG cost function. Disadvantages of the approach are the need for a fully connected communications network, the total operations performed over all the nodes are greater than for a centralized controller, and the approach is formulated for linear time-invariant systems. To investigate the feasibility of the decentralized approach to satellite formation flying, a simple centralized LQG design for a spacecraft orbit control problem is adapted to the decentralized framework. The simple design uses a fixed reference trajectory (an equatorial, Keplerian, circular orbit), and by appropriate choice of coordinates and measurements is formulated as a linear time-invariant system.

INTRODUCTION

A decentralized framework for linear-quadratic-Gaussian (LQG) control is investigated for applicability to autonomous satellite formations. For such problems, a solution that minimizes data transmission requirements has been given by Speyer [1]. In reference [1], the decentralized estimator was placed in an LQG control setting. Since then other decentralized control algorithms have been analyzed which consider one-step delayed-information sharing patterns. In reference [2], the decentralized LQG control is extended to the decentralized linear-exponential-Gaussian control which is related to deterministic $\mathcal{H}_\infty$ control synthesis. Other generalizations of reference [1] may be found in references [3] and [4], and in reference [5], reference [1] served as the basis for a fault-tolerant multi-sensor navigation architecture.

The decentralized LGQ framework is non-hierarchical and coordination by a central supervisor is not required. Detected failures degrade the system performance gracefully. Each node in the decentralized network processes only its own measurement data, in parallel with the other nodes. Although the total computational burden over the entire network is greater than it would be for a single, centralized controller, fewer computations are required locally at each node. Requirements for data transmission between nodes are limited to only the dimension of the control vector, at the cost of maintaining a local additional data vector. The data vector compresses all past measurement history from all the nodes into a single vector of the dimension of the state. The approach is optimal with respect to standard LQG cost function.

As literally formulated in reference [1], the approach is valid for linear time-invariant (LTI) systems only. As with the standard LQG problem, extension to linear time-varying (LTV) systems requires that each node propagate its filter covariance forward and controller Riccati matrix backward at each time step. Extension to non-linear systems can also be accomplished via linearization about a reference trajectory in the standard fashion, or linearization about the current state estimate as with the extended Kalman filter. Each of these extensions induces additional local processing and data transmission requirements, however.

To investigate the feasibility of the decentralized approach to satellite formation flying, an existing centralized LQG design for a single spacecraft orbit control problem is adapted to the decentralized framework. The existing design uses a fixed reference trajectory, and by appropriate choice of coordinates and simplified measurement modeling is formulated as a linear time-invariant system.

The remainder of this paper is organized as follows. The next two sections briefly describe the decentralized control approach of reference [1] and the simplified spacecraft orbit control LQG design, respectively. The next section presents and discusses results for a particular three satellite formation, and highlights a few issues relevant to this problem. The final section discusses the feasibility of extending this work to the kind of time-varying models required for use in an actual distributed spacecraft application.

BACKGROUND

Reference [1] considers the solution to the discrete[1] decentralized LQG control problem, described by the minimization of

$$J = \min_{\mathbf{u}_i^j} \mathrm{E}\left[\frac{1}{2}\sum_{i=1}^{N}\left\{\mathbf{x}_i^{\mathsf{T}}\mathbf{Q}_i\mathbf{x}_i + \sum_{j=1}^{K}(\mathbf{u}_i^j)^{\mathsf{T}}\mathbf{R}_i^j\mathbf{u}_i^j\right\}\right], \qquad j = 1,2,\ldots,K; i = 1,2,\ldots,N,$$

subject to the linear time invariant measurement update given by

$$\mathbf{y}_i^j = \mathbf{H}^j\mathbf{x}_i + \mathbf{v}_i^j,$$

and the linear time invariant state update given by

$$\mathbf{x}_{i+1} = \mathbf{\Phi}\mathbf{x}_i + \sum_{j=1}^{K}\mathbf{B}^j\mathbf{u}_i^j + \mathbf{w}_i,$$

where $\mathbf{x}_1 \sim N(\bar{\mathbf{x}}, \bar{\mathbf{P}})$, $\mathbf{w}_i \sim N(\mathbf{0}, \mathbf{W}_i\delta_{i\ell})$, $\mathbf{v}_i^j \sim N(\mathbf{0}, \mathbf{V}_i^j\delta_{i\ell})$, and where K is the number of nodes in the network, and N is the number of epochs over which the system operates. Figure 1 illustrates the approach taken by reference [1].

The solution is based on the decomposition of the state into $\mathbf{x}_i^{\mathcal{C}}$, that depends only on the control, and $\hat{\mathbf{x}}_i^{\mathcal{D}}$, that depends only on the incoming data. The global Riccati matrices $\mathbf{P}$ and $\mathbf{S}$ are computed off-line via

$$\begin{aligned}
\mathbf{P}_{i+1}^{-1} &= \mathbf{M}_{i+1}^{-1} + \sum_{j=1}^{K}(\mathbf{H}^j)^{\mathsf{T}}(\mathbf{V}_{i+1}^j)^{-1}\mathbf{H}^j \\
\mathbf{M}_{i+1} &= \mathbf{\Phi}\mathbf{P}_i\mathbf{\Phi}^{\mathsf{T}} + \mathbf{W}_i; \qquad \mathbf{M}_1 = \bar{\mathbf{P}} \\
\mathbf{S}_i &= \mathbf{\Phi}^{\mathsf{T}}\mathbf{S}_{i+1}\mathbf{\Phi} - \sum_{j=1}^{K}(\mathbf{L}_i^j)^{\mathsf{T}}(\mathbf{R}^j + (\mathbf{B}^j)^{\mathsf{T}}\mathbf{S}_{i+1}\mathbf{B}^j)\mathbf{L}_i^j + \mathbf{Q}_i, \quad \mathbf{S}_N = \mathbf{0},
\end{aligned}$$

where

$$\mathbf{L}_i^j = -(\mathbf{R}^j + (\mathbf{B}^j)^{\mathsf{T}}\mathbf{S}_{i+1}\mathbf{B}^j)^{-1}(\mathbf{B}^j)^{\mathsf{T}}\mathbf{S}_{i+1}.$$

The local filter covariance matrix, $\mathbf{P}_i^j$, is also computed offline via

$$\begin{aligned}
(\mathbf{P}_{i+1}^j)^{-1} &= (\mathbf{M}_{i+1}^j)^{-1} + (\mathbf{H}^j)^{\mathsf{T}}(\mathbf{V}_{i+1}^j)^{-1}\mathbf{H}^j \\
\mathbf{M}_{i+1}^j &= \mathbf{\Phi}\mathbf{P}_i^j\mathbf{\Phi}^{\mathsf{T}} + \mathbf{W}_i; \qquad \mathbf{M}_1 = \bar{\mathbf{P}}.
\end{aligned}$$

The vectors $\mathbf{h}_i^j$ are data-dependent vectors that efficiently compress non-local information. The vectors $\boldsymbol{\alpha}_i^{\ell j}$ are transmission vectors that have the dimensions of the control vectors.

Note that the only information that need be exchanged over the network are vectors that have the dimensions of the controls. The local control $\mathbf{u}_i^j$ cannot be computed until $\alpha_i^{j\ell}, \ell = 1,2,\ldots,j-1,j+1,\ldots,K$ that are received from the network at junction $\mathcal{B}$ have been computed at all the other nodes' junctions $\mathcal{C}$ (see Figure 1). If $\mathbf{B}^j$ are all the same, then $\mathbf{u}_i^j$ are all the same, so the sum at junction $\mathcal{A}$ does not require a network connection, and the $\mathbf{u}_i^j$ need not be exported to the network.

If the solution approach is used for decentralized control of a distributed satellite cluster, there are a few potential issues. Each node may be associated with only a partition of the global state. However, this is shown not to be a limitation in reference [3]. The system may be time-varying, so that the Riccati matrices become data-dependent, and may not be computed off-line. A terminal penalty function may be present in the cost function, J. Finally, it may not be possible to accommodate transmission of the $\alpha_i^{\ell j}$ and reception of the $\alpha_i^{j\ell}$ all during the current stage i.

[1] A continuous-time solution is also given.

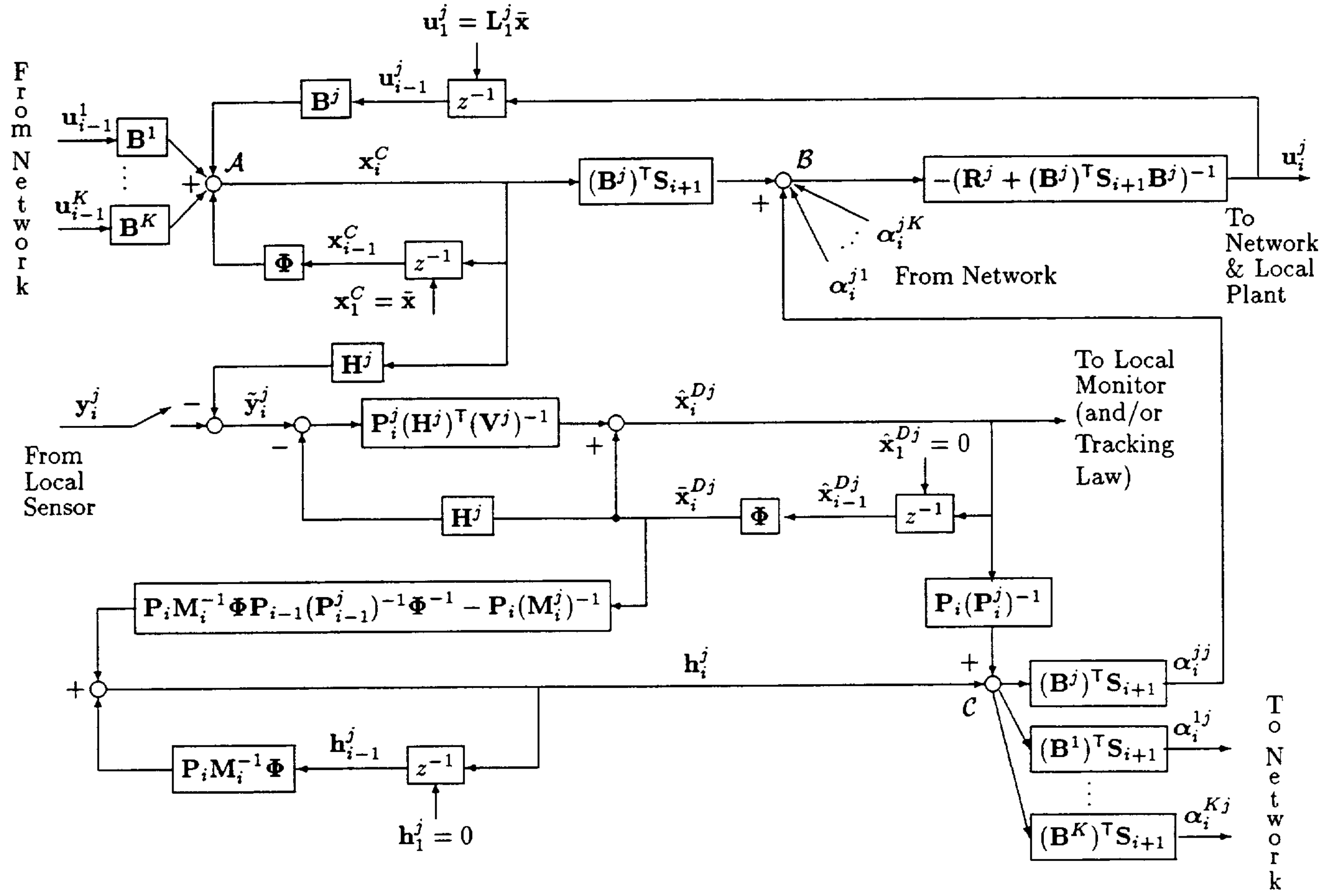


Figure 1: Block Diagram of Speyer's Solution for Decentralized LQG Control of a Linear Time-Invariant Plant

LQG DESIGN

Objectives

The time-varying linear-quadratic regulator (LQR) minimizes the criterion

$$J = \int_{t_1}^{t_N} \left[x(\tau)^T Q(\tau) x(\tau) + u(\tau)^T R(\tau) u(\tau) \right] d\tau + x(t_N)^T S_N x(t_N)$$

with respect to the linear, time-varying (LTV) system

$$\dot{x}(t) = A(t)x(t) + B(t)u(t); \quad x(t_1) = x_1. \tag{1}$$

The time-varying matrices $Q(\tau), R(\tau) : Q(\tau) \geq 0, R(\tau) > 0, \tau \in [t_1, t_N]$, act as penalty functions on the states, $x(t)$, and the controls, $u(t)$, respectively. Additionally, the matrix $S_N : S_N \geq 0$ forms a penalty on the final states. Because the weighting matrices in the LQR framework are arbitrary, this framework represents a class of optimal regulators. In order to choose the controller, the following additional performance metrics are defined:

$$\pi_p = \sup_{j \in \mathcal{J}_p} \|x_j(t)\|_\infty ; \quad \mathcal{J}_p \sim \text{position channels} \tag{2}$$

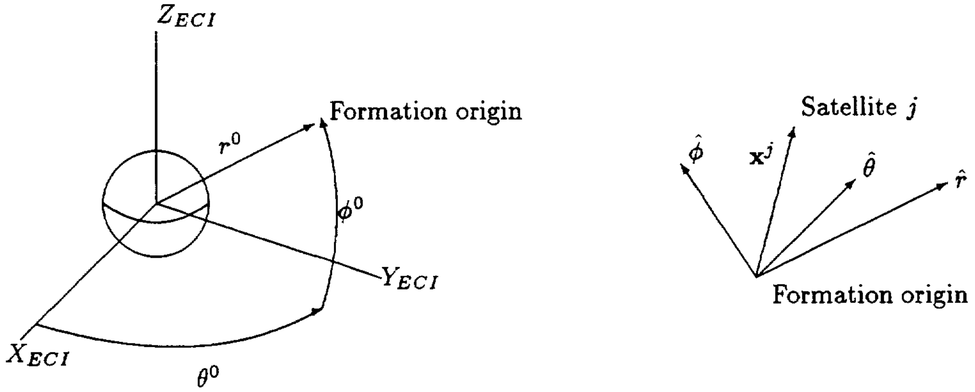
$$\pi_v = \sup_{j \in \mathcal{J}_v} \|x_j(t)\|_\infty ; \quad \mathcal{J}_v \sim \text{velocity channels} \tag{3}$$

$$\pi_u = \sup_{j \in \mathcal{J}_u} \frac{1}{N}\sqrt{\sum_i^N \mathbf{u}_j^2(t_i)}; \quad \mathcal{J}_u \sim \text{control channels, } N \sim \text{signal length} \tag{4}$$

The basis for the first two criteria, π_p and π_v, is to identify the controller design which generates the smallest maximum value of the states corresponding to position and velocity. The third criterion, π_u, represents the maximum among the mean square values of the controls, and is based on the total "control effort."

Linearization of System Dynamics

Before a controller can be designed, the dynamics of the distributed satellites' orbits must be expressed in the form of an LTV system, as specified by Eq. (1). Since these dynamics are represented by a high-order, nonlinear, nonconservative system of differential equations, appropriate simplifying assumptions must be used to achieve reductions in mathematical complexity. Then, linearization of the resulting simple non-linear equations about a reference orbit is employed to achieve a linear, time-invariant system, which forms a subclass of the systems represented by Eq. (1).



(a)

(b)

Figure 2: Problem Geometry: (a) Position of formation origin relative to Earth-Centered Inertial (ECI) frame; (b) Position of jth satellite relative to formation origin

The satellites are assumed to be orbiting the earth in a near-equatorial, near-circular orbit. Each satellite is further assumed to remain in the vicinity of a point, the *formation origin*, that orbits the earth in an equatorial, circular orbit. Its position is specified by the spherical coordinates r^0, θ^0, and ϕ^0, defined in Figure 2(a). It is also assumed that each satellite has small thrusters that it can use to apply accelerations u_r, u_θ, and u_ϕ in the $\hat{r}$, $\hat{\theta}$, and $\hat{\phi}$ (radial, downtrack, and crosstrack) directions, defined in Figure 2(b). A state vector and control vector are chosen for each satellite $j = 1, 2, \ldots, K$ (for $j = 0$, the state vector refers to the formation origin) as follows:

$$\mathbf{X}^j = \left[r^j,\ \dot{r}^j,\ \theta^j,\ \dot{\theta}^j,\ \phi^j,\ \dot{\phi}^j\right]^{\mathsf{T}}, \qquad j = 0, 1, 2, \ldots, K$$

$$\mathbf{U} = \left[u_r^j,\ u_\theta^j,\ u_\phi^j\right]^{\mathsf{T}}, \qquad j = 1, 2, \ldots, K$$

Each satellite's position relative to the formation origin is therefore given by

$$\mathbf{x}^j = \mathbf{X}^j - \mathbf{X}^0,$$

as Figure 2(b) depicts.

In terms of the states chosen above, the satellite dynamics may be expressed as a first-order, non-linear, autonomous, vector differential equation of the form

$$\dot{\mathbf{X}}^j = \mathbf{f}(\mathbf{X}^j, \mathbf{U}^j),$$

where

$$\mathbf{f}(\mathbf{X}^j,\mathbf{U}^j) = \left.\begin{bmatrix} \dot{r} \\ -\frac{\mu}{r^2} + r\dot{\phi}^2 + r\dot{\theta}^2\cos^2\phi + \frac{u_r}{m} \\ \dot{\theta} \\ -2\dot{r}\dot{\theta}/r + 2\dot{\phi}\dot{\theta}\tan\phi + \frac{u_\theta}{mr} \\ \dot{\phi} \\ -2\dot{r}\dot{\phi}/r - \dot{\theta}^2\cos\phi\sin\phi + \frac{u_\phi}{mr} \end{bmatrix}\right|_j . \tag{5}$$

In order to linearize the system of Eq. (5), the partial derivatives of **f** with respect to the states and controls are required. To simplify the notation, the parameter n, known as the *mean motion*, or *orb-rate*, which is defined by

$$n = \sqrt{\frac{\mu}{r^3}}, \tag{6}$$

will be employed. Now, defining the *state sensitivity matrix* as

$$\mathbf{A}^j = \frac{\partial\mathbf{f}(\mathbf{X}^j,\mathbf{U}^j)}{\partial\mathbf{X}^j},$$

then

$$\mathbf{A}^j = \left.\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ (\dot{\phi}^2 + 2n^2 + \dot{\theta}^2\cos^2\phi) & 0 & 0 & 2r\dot{\theta}\cos^2\phi & -2r\dot{\theta}^2\cos\phi\sin\phi & 2r\dot{\phi} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 2\dot{r}\dot{\theta}/r^2 & -2\dot{\theta}/r & 0 & 2(\dot{\phi}\tan\phi - \dot{r}/r) & 2\dot{\phi}\dot{\theta}\sec^2\phi & 2\dot{\theta}\tan\phi \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 2\dot{r}\dot{\phi}/r^2 & -2\dot{\phi}/r & 0 & -2\dot{\theta}\cos\phi\sin\phi & \dot{\theta}^2\cos 2\phi & -2\dot{r}/r \end{bmatrix}\right|_j \tag{7}$$

Defining the reference state trajectory,

$$\mathbf{X}_* = [r_*,\ 0,\ n_* t,\ n_*,\ 0,\ 0]^{\mathsf{T}},$$

and the reference control $\mathbf{U}_* = [0,0,0]^{\mathsf{T}}$, where r_* is the reference radius, and n_* is the reference orb-rate, defined in terms of r_* per Eq. (6), a great deal of simplification results:

$$\begin{aligned} \mathbf{A}_* &= \left.\frac{\partial\mathbf{f}(\mathbf{X},\mathbf{U})}{\partial\mathbf{X}}\right|_{\mathbf{X}_*,\mathbf{U}_*} \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 3n_*^2 & 0 & 0 & 2n_* r_* & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -2n_*/r_* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -n_*^2 & 0 \end{bmatrix}. \end{aligned}$$

If the state vector is now redefined as

$$\tilde{\mathbf{X}}^j = \left[r^j,\ \dot{r}^j,\ r_*\theta^j,\ r_*\dot{\theta}^j,\ r_*\phi^j,\ r_*\dot{\phi}^j\right]^{\mathsf{T}}$$

then the corresponding matrix $\tilde{A}_*$ is

$$\tilde{\mathbf{A}}_* = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 3n_*^2 & 0 & 0 & 2n_* & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -2n_* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -n_*^2 & 0 \end{bmatrix}.$$

Scaling the state vector in this fashion is motivated by computational issues. For almost any convenient set of units, r_* is apt to be quite large compared with n_*, which will cause wide disparities among the sizes of the elements of A_*. In addition to rectifying this problem, the chosen scaling has the cosmetic feature of expressing all the states in units of distance and time.

Finally, the *control sensitivity matrix* is

$$\mathbf{B}^j = \frac{\partial \mathbf{f}(\mathbf{X}^j, \mathbf{U}^j)}{\partial \mathbf{U}^j},$$

so that, after evaluating the partial derivatives on the reference trajectory, and applying the same scaling given above,

$$\tilde{\mathbf{B}}_* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that these assumptions have led to $\tilde{\mathbf{A}}_*$ and $\tilde{\mathbf{B}}_*$ that are the same for all nodes. Henceforth, the $\tilde{}$ and subscript $*$ notation will be dropped, and $\mathbf{A}$ and $\mathbf{B}$ will be understood to refer to $\tilde{\mathbf{A}}_*$ and $\tilde{\mathbf{B}}_*$, respectively.

Now deviations from the reference trajectory, defined by $\mathbf{x}^j = \mathbf{X}^j - \mathbf{X}_*$ and $\mathbf{u} = \mathbf{U}^j - \mathbf{U}_*$, may be written as a linear system as follows:

$$\dot{\mathbf{x}}^j = \mathbf{A}\mathbf{x}^j + \mathbf{B}\mathbf{u}^j. \tag{8}$$

This has the form of the linear dynamic system required for the LQR design (cf. Eq. (1).) Note that the $\mathbf{x}^j$ defined above is consistent with the definition in Figure 2(b), since the formation origin is assumed to follow the reference trajectory.

Discretization

To discretize this system, first consider the solution of the unforced system, $\dot{\mathbf{x}}^j(t) = \mathbf{A}\mathbf{x}^j(t)$:

$$\mathbf{x}^j(t) = \mathbf{\Phi}(t, t_1)\mathbf{x}^j(t_1),$$

where $\mathbf{\Phi}(t, t_1)$ is the state transition matrix. It can be shown (e.g. see Kaplan [6]) that $\mathbf{\Phi}(t, t_1) = \mathbf{\Phi}(n, \Delta t)$ for the two-body satellite problem, where

$$\mathbf{\Phi}(n, \Delta t) = \begin{bmatrix} \mathbf{\Phi}_{r,\theta}(n, \Delta t) & 0 \\ 0 & \mathbf{\Phi}_{\phi}(n, \Delta t) \end{bmatrix},$$

$$\mathbf{\Phi}_{r,\theta}(n, \Delta t) = \begin{bmatrix} 4 - 3\cos n\Delta t & \frac{1}{n}\sin n\Delta t & 0 & \frac{2}{n}(1 - \cos n\Delta t) \\ 3n \sin n\Delta t & \cos n\Delta t & 0 & 2\sin n\Delta t \\ 6(\sin n\Delta t - 1) & \frac{2}{n}(\cos n\Delta t - 1) & 1 & \frac{4}{n}\sin n\Delta t - 3t \\ 6n(\cos n\Delta t - 1) & -2\sin n\Delta t & 0 & 4\cos n\Delta t - 3 \end{bmatrix},$$

and

$$\mathbf{\Phi}_{\phi}(n, \Delta t) = \begin{bmatrix} \cos n\Delta t & \frac{1}{n}\sin n\Delta t \\ -n \sin n\Delta t & \cos n\Delta t \end{bmatrix}.$$

Here n is the orb-rate, defined previously, and $\Delta t = t - t_1$. Thus, once n and Δt are specified, $\mathbf{\Phi}$ is a constant matrix.

Using this unforced solution, the discrete form of Eq. (8) may be written as

$$\mathbf{x}^j_{i+1} = \mathbf{\Phi}(n_*, t_{i+1} - t_i)\mathbf{x}^j_i + \mathbf{\Lambda}\mathbf{u}^j_i \tag{9}$$

where

$$\mathbf{\Lambda}\mathbf{u}^j_i = \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(n_*, \Delta\tau)\mathbf{B}\mathbf{u}^j(\tau)d\tau,$$

To evaluate the integral, model $\mathbf{u}^j(t)$ as

$$\mathbf{u}^j(\tau) = \Delta\mathbf{v}_i^j \delta(\tau - t_i),$$

where $\Delta\mathbf{v}_i^j$ is an impulsive velocity change assumed to occur at time t_i.

The system model that has been derived is linear time-invariant. Therefore, as in reference [1], the Riccati matrices $\mathbf{S}_i$ and control gains $\mathbf{L}_i^j$ may be computed off-line. This would not likely be true for realistic formation flying applications, especially those involving highly elliptic orbits.

Tracking Desired Local Trajectories

The LQR framework provides a regulator, that is, a controller that drives state deviations to zero. For the distributed satellite control problem, one actually wishes to have each of the nodes track a desired local trajectory relative to the formation origin, denoted $\mathbf{x}_d^j(t_i)$. Thus, the objective is to drive $\mathbf{x}^j(t_i) - \mathbf{x}_d^j(t_i)$ to zero. Reference [7] shows that if

$$\dot{\mathbf{x}}_d(t_i) = \mathbf{A}\mathbf{x}_d(t_i),$$

then the control law

$$\mathbf{u}_i = \mathbf{L}_i[\mathbf{x}(t_i) - \mathbf{x}_d(t_i)],$$

where $\mathbf{L}_i$ is the LQ gain matrix, will achieve the desired objective[2].

In terms of the satellite control problem, this can be interpreted as requiring that the desired local trajectory be a free orbital trajectory. Several recent works (e.g., [8], [9], [10], [11]) have shown that for formation keeping to be economically feasible, the desired local trajectories should be free orbital trajectories, as nearly as possible. Use of the standard LQR framework imposes the necessity of this desirable condition.

Discrete Linear Kalman Filter Design

According to the certainty equivalence principle, the LQR design presented above is the optimal controller design, if the deviations from the reference trajectory result from Gaussian stochastic processes. However, to implement this design as an optimal control law, one must use an estimate of the state deviations that is based on observations of these random processes. Such an estimate is provided by the Kalman filter. Although in the satellite problem considered here, the state deviations do not result from stochastic processes, but rather from unmodeled dynamics, the implementation strategy suggested by the certainty equivalence principle will be employed[3].

Propagation Model

Assuming that the state deviations are indeed governed by the linear system used for the controller design, driven by white Gaussian noise, leads to the continuous model

$$\dot{\mathbf{x}}^j(t) = \mathbf{A}\mathbf{x}^j(t) + \mathbf{B}\mathbf{u}^j(t) + \mathbf{w}^j(t), \tag{10}$$

with $\mathbf{x}^j(t_1) = \bar{\mathbf{x}}^j$, $\mathbf{P}(t_1) = \mathrm{E}[(\bar{\mathbf{x}}^j - \mathrm{E}[\bar{\mathbf{x}}^j])(\bar{\mathbf{x}}^j - \mathrm{E}[\bar{\mathbf{x}}^j])^{\mathrm{T}}]$, $\mathbf{u}^j(t)$ specified, and $\mathrm{E}[\mathbf{w}^j(t)] = \mathbf{0}$, $\mathrm{E}[\mathbf{w}^j(t)(\mathbf{w}^j)^{\mathrm{T}}(\tau)] = \mathbf{W}^j\delta(t-\tau)$.

Using Eq. (9), the discrete form of Eq. (10) may be written as

$$\mathbf{x}_{i+1}^j = \mathbf{\Phi}\mathbf{x}_i^j + \mathbf{\Lambda}\mathbf{u}_i^j + \mathbf{w}_i^j, \tag{11}$$

where

$$\mathbf{w}_i^j = \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(\Delta\tau)\mathbf{w}^j(\tau)d\tau,$$

[2] If $\mathbf{x}_d(t_i)$ does not satisfy this condition, then the LQ cost function must be converted into an equivalent disturbance rejection problem [7], where the disturbance is $\mathbf{d}(t) = \mathbf{A}\mathbf{x}_d(t) - \dot{\mathbf{x}}_d(t)$.

[3] If viewing the unmodeled dynamics as Gaussian noise processes seems unreasonable, consider that it seems entirely plausible that some combination of (probably very many) distribution functions would yield a good approximation to these unmodeled dynamics. Then, in light of the central limit theorem, which states that combinations of many variously distributed stochastic processes tend toward a Gaussian stochastic process, the assumption seems valid.

with

$$\mathrm{E}\Big[\mathbf{w}_i^j(\mathbf{w}_i^j)^{\mathsf{T}}\Big] = \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(\Delta\tau)\mathbf{W}^j\mathbf{\Phi}^{\mathsf{T}}(\Delta\tau)d\tau, \qquad \mathrm{E}\Big[\mathbf{w}_i^j(\mathbf{w}_\ell^j)^{\mathsf{T}}\Big] = \mathbf{0}.$$

To evaluate this integral, a simplification is introduced. *For purposes of evaluating the process noise integral only*, the state is assumed to propagate according to $\mathbf{x}_{i+1}^j = \mathbf{x}_i^j + \dot{\mathbf{x}}_i^j\Delta t$. This results in an approximate state transition matrix,

$$\tilde{\mathbf{\Phi}}(\Delta t) = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Next, consider that since the state noise is used to account for unmodeled forces, it typically has non-zeros entries only for acceleration channels. If one further restricts the state noise to be uncorrelated between the various acceleration channels, $\mathbf{W}$ then has a diagonal structure, e.g.

$$\mathbf{W}^j = \mathrm{diag}(0, w_{\ddot{r}}, 0, w_{r\ddot{\theta}}, 0, w_{r\ddot{\phi}}),$$

which is here restricted to be the same for all nodes for simplicity, but this is not necessary. Then,

$$\mathbf{W}_i^j = \mathrm{E}\Big[\mathbf{w}_i^j(\mathbf{w}_i^j)^{\mathsf{T}}\Big] = \int_{t_i}^{t_{i+1}} \tilde{\mathbf{\Phi}}(\Delta\tau)\mathbf{W}^j\tilde{\mathbf{\Phi}}^{\mathsf{T}}(\Delta\tau)d\tau,$$

so that

$$\mathbf{W}_i^j = \begin{bmatrix} w_{\ddot{r}}\Delta t^3/3 & w_{\ddot{r}}\Delta t^2/2 & 0 & 0 & 0 & 0 \\ w_{\ddot{r}}\Delta t^2/2 & w_{\ddot{r}}\Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & w_{r\ddot{\theta}}\Delta t^3/3 & w_{r\ddot{\theta}}\Delta t^2/2 & 0 & 0 \\ 0 & 0 & w_{r\ddot{\theta}}\Delta t^2/2 & w_{r\ddot{\theta}}\Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{r\ddot{\phi}}\Delta t^3/3 & w_{r\ddot{\phi}}\Delta t^2/2 \\ 0 & 0 & 0 & 0 & w_{r\ddot{\phi}}\Delta t^2/2 & w_{r\ddot{\phi}}\Delta t \end{bmatrix}.$$

Measurement Model

Since the purpose of this work is to begin the process of evaluating the feasibility of the decentralized control approach to distributed satellites, a very simple linear time-invariant measurement model corresponding to noise-corrupted measurements of geocentric radius, longitude, and latitude is assumed[4]. The assumed measurement model is

$$\mathbf{Y}_i^j = \mathbf{H}^j\mathbf{X}_i^j + \mathbf{v}_i^j,$$

with $\mathrm{E}\Big[\mathbf{v}_i^j\Big] = \mathbf{0}$, $\mathrm{E}\Big[\mathbf{v}_i^j(\mathbf{v}_\ell^j)^{\mathsf{T}}\Big] = \mathbf{V}^j\delta_{i\ell}$, where

$$\mathbf{H}^j = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/r_o & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/r_o & 0 \end{bmatrix},$$

which is the same for all nodes. Recall that $\mathbf{X}^j(t)$ is the nonlinear state vector, whereas the controller operates on the state deviations, $\mathbf{x}^j(t)$. If the filter operates on the "measurement deviations," i.e.,

$$\begin{aligned} \mathbf{y}_i^j &= \mathbf{H}^j\mathbf{X}_i^j + \mathbf{v}_i^j - \mathbf{H}^j\mathbf{X}_*(t_i) \\ &= \mathbf{H}^j\mathbf{x}_i^j + \mathbf{v}_i^j, \end{aligned}$$

then a linearized Kalman filter design results.

[4] This is quite an unrealistic assumption, and in fact the primary manner in which time-varying systems enter the distributed satellite problem is through the measurements.

Modifications to LQ Design to Accommodate Decentralized Control Framework

For the decentralized control solution approach of reference [1], a single state space is assume to be shared by all the nodes. Reference [3] shows that it is sufficient if each of the local state spaces may be linearly extracted from the global state space, e.g. $\mathbf{x}^j = \mathbf{T}\mathbf{x}$. In the context of the distributed satellite control problem, this condition is satisfied if the global state $\mathbf{x}$ is defined to be the column vector consisting of all the local states $\mathbf{x}^j$ arranged columnwise, e.g.

$$\mathbf{x}^{\mathsf{T}} = \left[(\mathbf{x}^1)^{\mathsf{T}}, (\mathbf{x}^2)^{\mathsf{T}}, \ldots, (\mathbf{x}^K)^{\mathsf{T}}\right].$$

The global state sensitivity matrix is then a block diagonal matrix with the (identical) local state sensitivity matrices along its diagonal.

The solution approach of reference [1] also requires that the local states be decomposed into $\mathbf{x}_i^{Cj}$, that depend only on the control, and $\hat{\mathbf{x}}_i^{Dj}$, that depend only on the incoming data. The Kalman filter design presented above operates on the entire state, as would all existing filter designs that might be used. To accommodate the need for $\hat{\mathbf{x}}_i^{Dj}$ in computing the control, $\mathbf{x}_i^{Cj}$ was computed external to the filter, and subtracted from the whole state output of the filter to get $\hat{\mathbf{x}}_i^{Dj}$.

MISSION SIMULATION AND RESULTS

To begin to study the feasibility of the decentralized LQG control strategy described above to satellite formation flying, an example of a distributed satellite mission is considered. The example mission selected, which Figure 3 illustrates, could be of interest for making distributed sparse radar observations of earth-based targets, since the projection of the satellites' motion relative to one another onto a locally level plane is circular. Thus, the satellites appear to maintain constant baselines relative to an earth-based target. Perturbations to the two-body motion must be compensated in order to maintain desired configuration.

Since the purpose of this work is feasibility assessment only, the simulation model is identical to the design model used for the LQG controller, and all perturbations are Gaussian white noise processes. The commanded maneuvers are without execution error, and the measurements are bias-free. To complete the LQG design, the free parameters must be specified. After a few design iterations, the values indicated in Table 1 were selected, based on having generally adequate performance in terms of Eqs. (2)-(4). The reference orbit radius and the simulation time step are $r_* = 7178$ km and $\Delta t = 1$ min., respectively. Note that the measurements and maneuvers are assumed to be synchronized to occur on the simulation time steps.

Table 1: LQG Design Parameters

Parameter	Nomenclature	Value(s)
$\mathbf{Q}$	State Penalty Matrix	I
$\mathbf{R}^j$	Control Penalty Matrix	$\mathrm{diag}(1.75e8, 1e8, 1e8)$
$\mathbf{S}_N$	Final State Penalty Matrix	$\mathbf{0}$
$\mathbf{V}^j$	Measurement Noise Covariance	$\mathrm{diag}(50, 25, 25)$
$w_{\ddot{r}}, w_{\ddot{r\theta}}, w_{\ddot{r\phi}}$	Process Noise Spectral Densities	$9.81e-4$
$\mathbf{P}^j(t_o)$	Initial Covariance	$\sigma_r = \sigma_{r\theta} = \sigma_{r\phi} = 100$, $\sigma_{\dot{r}} = \sigma_{\dot{r\theta}} = \sigma_{\dot{r\phi}} = 0.1$, $\rho_{r,(\dot{r\theta})} = \rho_{(r\theta),\dot{r}} = -.9$, all other elements zero

Note that the use of a penalty function on the states along the path has been used (since $\mathbf{Q} \neq \mathbf{0}$), and a terminal penalty function has not used (since $\mathbf{S}_N = \mathbf{0}$). It has been more common when developing spacecraft maneuvers to use the opposite strategy, e.g. Lambert targeting and the Hohmann transfer, which provide only for meeting final constraints, and do not penalize deviations along the path from a desired trajectory. Such approaches might not be adequate for a tight satellite formation however, where collision and near-miss avoidance becomes an issue. In such a scenario, it could be essential that the vehicles remain close to their pre-planned trajectories, rather than take arbitrary Keplerian paths between current and desired final states.

During observation periods, it is likely that maneuvering may be inhibited for various reasons such as minimizing vibrational disturbances onboard the vehicles, preventing corruption of instruments from thruster plumes, etc. It is

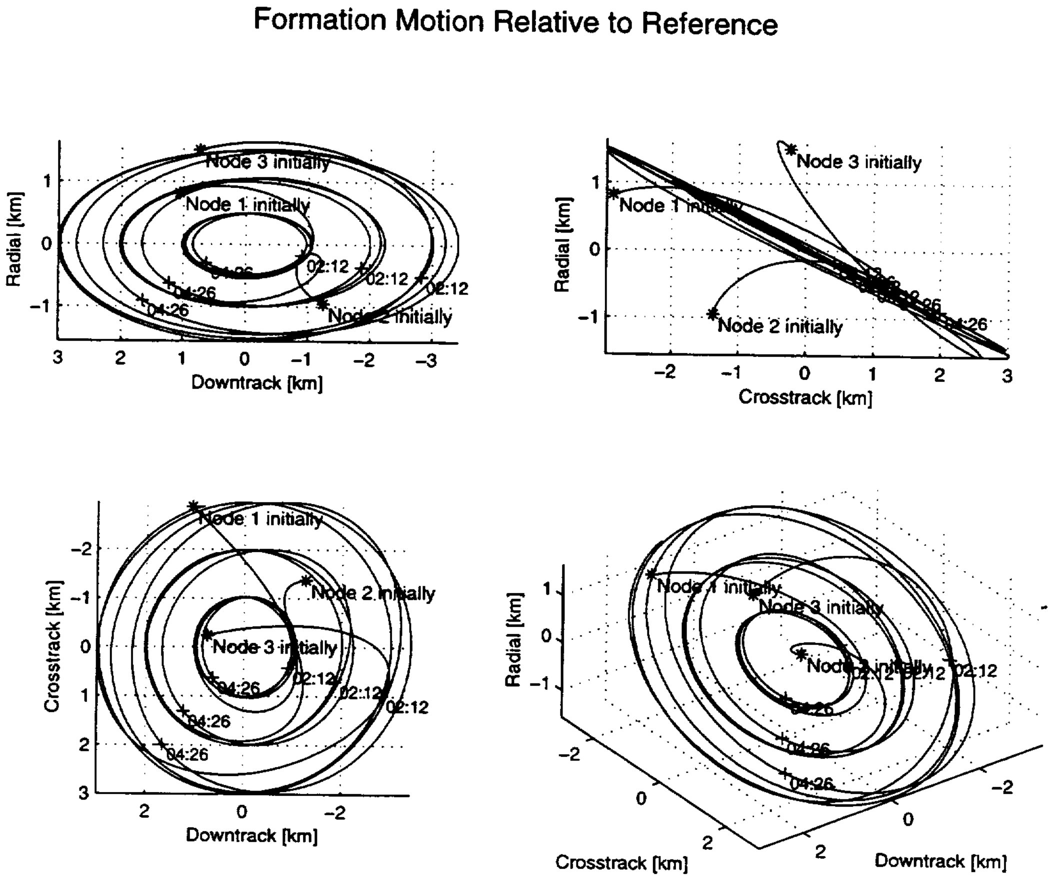


Figure 3: Controlled motion of distributed satellites. Upper left—in-plane motion cycles clockwise; upper right—out-of-plane motion constrained to inclined plane; lower left—motion viewed from zenith is circular; lower right—isometric view of motion

also possible that small, high specific-impulse thrusters would be used that would be pulsed at high rates, applying a small velocity increment to the vehicles at each pulse interval in order to accomplish maneuvers. Based on these assumptions, the simulation consists of three distinct time intervals: (1) an initial maneuver window lasting a little over one orbit (100 minutes) during which the satellites are initially driven onto the desired local reference trajectories, (2) an observation period lasting approximately two orbits during which no maneuvers are performed and perturbations tend to drive the satellites away from their desired local references, and (3) a final maneuver window lasting a little over one orbit (100 minutes) during which the satellites are driven back to their desired local references.

This strategy and its effects on the satellites is most clearly evident in Figure 4. The lowest subplot shows the commanded velocity increments, which are clearly zero during the observation period from about 01:40 to 05:00 hours. During this time, the position and velocity deviations that the upper subplots depict gradually build up, but are driven back to zero during the final maneuver window. This plot is typical of the other nodes' performance, which must be ommitted here for brevity. For completeness, the performance of the navigation system is also illustrated by Figure 5. The filter's estimation errors remain within the filter's expected error region, and have the signature of uncorrelated errors. The filter's performance is not affected by the maneuvering that occurs during the first 01:40 hours and from 05:00 to 06:40 hours.

CONCLUSIONS

The decentralized control approach of reference [1] has been successfully applied to a simplified version of a distributed satellite contol problem. This constitutes a first step in the assessment of the feasibility of this method for use in satellite formation flying missions. Based on the promising results acheived with this phase of the feasibility assessment, further

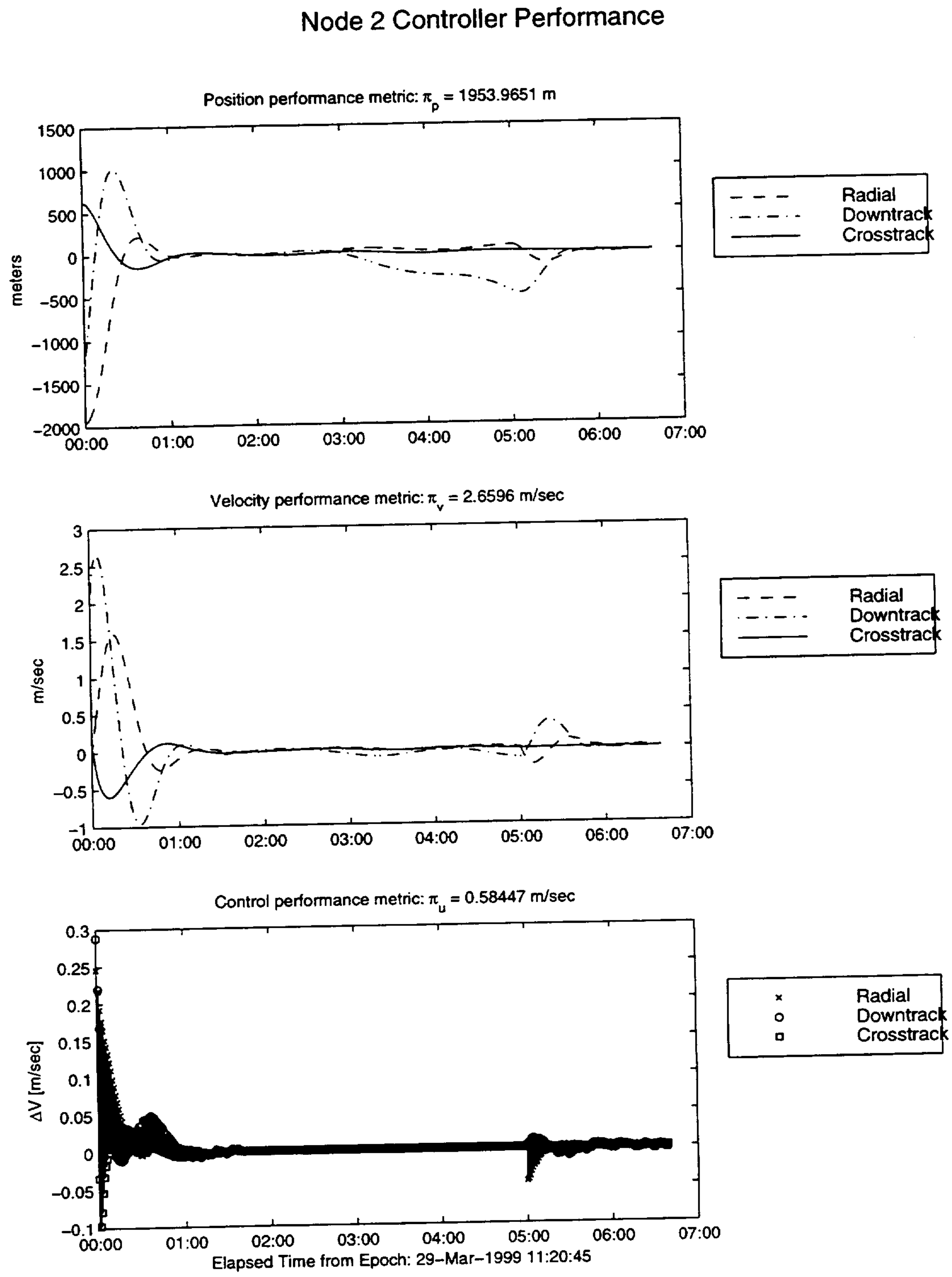


Figure 4: Controller performance (node 2), with maneuver windows during orbits 1 and 4. Upper and middle subplots—position and velocity deviations from desired local trajectory. Lowest subplot—velocity increments commanded by control law.

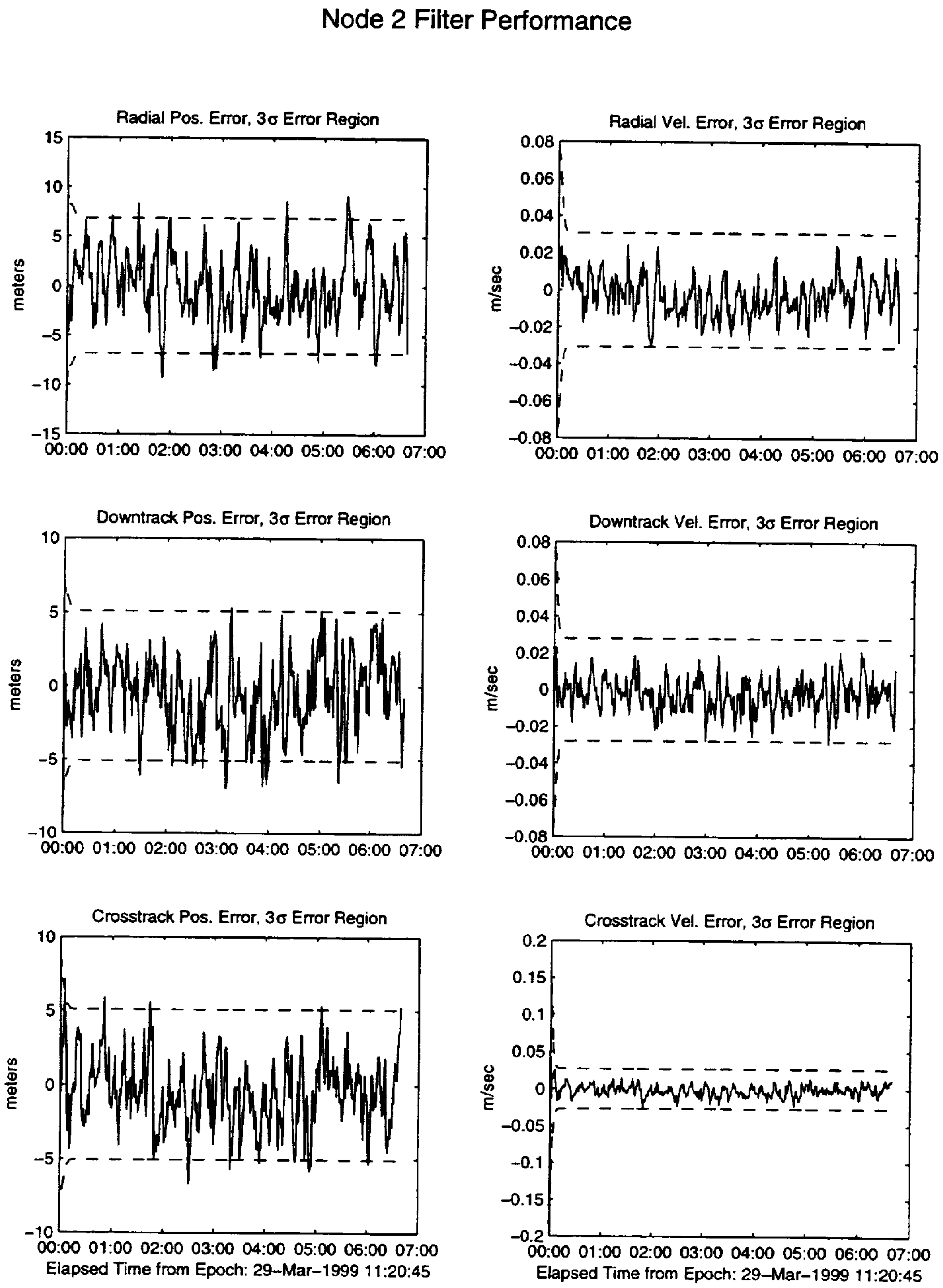


Figure 5: Filter performance for node 2

study of the implications of relaxing the simplifying assumptions used here are expected to be pursued.

Note that in the simplified formulation used in this paper, the local measurements are functions only of the local state. Also, the desired local trajectories are specified relative only to the formation center, and not with respect to the other nodes. Therefore, there is no advantage *from the control systems point of view* for information sharing. However, mission objectives such as coordination of observations or fleet reconfiguration could require each node to have knowledge of the states of all the other nodes. The decentralized framework provides an efficient means for communicating this information with minimal data transmission requirements. Also, in a realistic distributed spacecraft application, it is likely that relative measurements among the nodes would be utilized. In this scenario, the optimal local filter and therefore the optimal control requires knowledge of the states of all the nodes involved in the relative measurement.

Note also that the effects of command and data handling system and communications channel noise and latency must be considered. For example, it is likely that the measurement devices and/or their associated command and data handling systems may provide asynchronous and/or non-simultaneous measurement data. It is also probable that transmission of the data between nodes could be delayed one or more sampling periods and/or interrupted for extended intervals. Furthermore, since it is not possible to create a noise-free communications channel, errors in the transmitted data will be introduced at the level of the network, in addition to errors introduced at the measurement and state transition levels. An additional difficulty is that, in general, nonlinearities in the dynamics and measurements of the distributed satellite control problem require the use of ad-hoc procedures such as the extended Kalman filter.

References

[1] Speyer, J. L., "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem," *IEEE Trans. Automat. Contr.*, vol. AC-24(2), 1979, pp. 266-269.

[2] Fan, C.-H., J. L. Speyer, and C. R. Jaensch, "Centralized and Decentralized Solutions of the Linear-Exponential-Gaussian Problem," *IEEE Trans. Automat. Contr.*, vol. AC-39(10), 1994.

[3] Willsky, A. S., et al., "Combining and Updating of Local Estimates and Regional Maps Along Sets of One-Dimensional Tracks," *IEEE Trans. Automat. Contr.*, vol. AC-27(4), 1982, pp. 799-813.

[4] Levy, B. C., et al., "A Scattering Framework for Decentralized Estimation Problems," *Automatica*, vol. 19(4), 1983, pp. 373-387.

[5] Kerr, T., "Decentralizeed Filtering and Redundancy Management for Multisensor Navigation," *IEEE Trans. Aerosp. Electr. Sys.*, vol. AES-23(1), 1987, pp. 83-119. Also, Kerr, T., "Corrections to 'Decentralizeed Filtering and Redundancy Management for Multisensor Navigation,"' *IEEE Trans. Aerosp. Electr. Sys.*, vol. AES-23(3), 1987, p. 412.

[6] Kaplan, M. H., *Modern Spacecraft Dynamics and Control*, Wiley, New York, 1976, pp. 109-115.

[7] Dorato, P., Abdallah, C., and Cerone, V., *Linear-Quadratic Control: An Introduction*, Prentice-Hall, Englewood Cliffs, NJ, 1995, pp. 44-50.

[8] Sabol, C., Burns, R., and McLaughlin, C. A., "Satellite Formation Flying Design and Evolution," Paper AAS 99-121, *AAS/AIAA Space Flight Mechanics Meeting*, Feb. 7-10, 1999, Breckenridge, CO.

[9] Kong, E. M. C., Miller, D. W., and Sedwick, R. J., "Exploiting Orbital Dynamics for Aperture Synthesis Using Distributed Satellite Systems: Applications to a Visible Earth Imager System," Paper AAS 99-122, *AAS/AIAA Space Flight Mechanics Meeting*, Feb. 7-10, 1999, Breckenridge, CO.

[10] Sedwick, R. J., Miller, D. W., and Kong, E. M. C., "Mitigation of Differential Perturbations in Clusters of Formation Flying Satellites," Paper AAS 99-124, *AAS/AIAA Space Flight Mechanics Meeting*, Feb. 7-10, 1999, Breckenridge, CO.

[11] Tan, Z., Bainum, P. M., and Strong, A., "A Strategy for Maintaining Distance Between Satellites in an Orbiting Constellation," Paper AAS 99-125, *AAS/AIAA Space Flight Mechanics Meeting*, Feb. 7-10, 1999, Breckenridge, CO.

AUTONOMOUS FORMATION FLYING FROM THE GROUND TO FLIGHT

Keith B. Chapman (chapman@ai-solutions.com)
Gregory T. Dell (dell@ai-solutions.com)
Duane L. Rosenberg (rosenberg@ai-solutions.com)
AI Solutions, Inc.
10001 Derekwood Lane, Suite 215
Lanham, MD 20706
(301) 306-1756
FAX: (301) 306-1754

John Bristow (john.bristow@gsfc.nasa.gov)
National Aeronautics and Space Administration
Goddard Space Flight Center, Code 572
Greenbelt, MD 20771
(301) 286-3647
FAX: (301) 286-0369

ABSTRACT

The cost of on-orbit operations remains a significant and increasingly visible concern in the support of satellite missions. Headway has been made in automating some ground operations; however, increased mission complexity and more precise orbital constraints have compelled continuing human involvement in mission design and maneuver planning operations. AI Solutions, Inc. in cooperation with the National Aeronautics and Space Administration's (NASA) Goddard Space Flight Center (GSFC) has tackled these more complex problems through the development of AutoCon™ as a tool for an automated solution. NASA is using AutoCon™ to automate the maneuver planning for the Earth Orbiter –1 (EO-1) mission.

AutoCon™ was developed originally as a ground system tool. The EO-1 mission will be using a scaled version of AutoCon™ on-board the EO-1 satellite to command orbit adjustment maneuvers. The flight version of AutoCon™ plans maneuvers based on formation flying algorithms developed by GSFC, JPL, and other industry partners. In its fully autonomous mode, an AutoCon™ planned maneuver will be executed on-board the satellite without intervention from the ground.

This paper describes how AutoCon™ automates maneuver planning for the formation flying constraints of the EO-1 mission. AutoCon™ was modified in a number of ways to automate the maneuver planning on-board the satellite. This paper describes how the interface and functionality of AutoCon™ were modified to support the on-board system. A significant component of this modification was the implementation of a data smoother, based on a Kalman filter, that ensures that the spacecraft states estimated by an on-board GPS receiver

are as accurate as possible for maneuver planning. This paper also presents the methodology used to scale the AutoCon™ functionality to fit and execute on the flight hardware.

This paper also presents the modes built into the system that allow the incremental phasing in of autonomy. New technologies for autonomous operations are usually received with significant, and probably appropriate, trepidation. A number of safeguards have been designed in both AutoCon™ and the interfacing systems to alleviate the potential of mission-impacting anomalies from the on-board autonomous system. This paper describes the error checking, input data integrity validation, and limits set on maneuvers in AutoCon™ and the on-board system.

INTRODUCTION

The cost of on-orbit operations remains a significant and increasingly visible concern in the support of satellite missions. To reduce mission planning and on-orbit operations costs, and to enable better science return, NASA GSFC has teamed with AI Solutions, Inc. to develop AutoCon™, an automated maneuver planning tool.

NASA is using AutoCon™ as an experiment in on-board automation and formation control. AutoCon™ will automate the maneuver planning for the New Millennium Program (NMP) Earth Orbiter (EO) -1 mission. This paper discusses the conversion of the interactive GUI AutoCon™ ground system to a flight system.

BACKGROUND

AutoCon™

More space missions are requiring maneuvers so frequently that the mission itself would be cost prohibitive without automation. In addition, analysts are demanding friendlier, interactive and graphical tools that are very powerful and intelligent. AutoCon™, a ground-based mission planning tool, was developed to satisfy these needs. Ground AutoCon™ or AutoCon™-G includes a user friendly GUI, graphical plots (including 2D and 3D world maps) and report generation. AutoCon™ uses fuzzy logic to resolve multiple conflicting constraints, and plan maneuvers with little or no human interaction. Fuzzy logic can be used to control mission planning through a rule-based scheme. For example, a maneuver might be planned if the spacecraft is near a certain point in the orbit such as apogee or a descending node, and the time at the control center is around midday. Mission and instrument constraint rules can also be incorporated into a flexible maneuver-planning scenario. Hedges such as *almost*, *somewhat*, and *very* allow the rules to be adjusted easily. AutoCon™ is also scalable in such a way as to be put on-board the spacecraft and provide completely autonomous control including formation flying.

The on-board flight version of AutoCon™, or AutoCon™-F, developed for EO-1 consists of a subset of the ground based AutoCon™ application with a flight software interface. Figure 1 shows the functionality included in the flight and ground versions of AutoCon™. The flight interface, or Enhanced Formation Flying (EFF) software, interfaces directly with the Command and Data Handling (C&DH) system to retrieve all AutoCon™ required data,

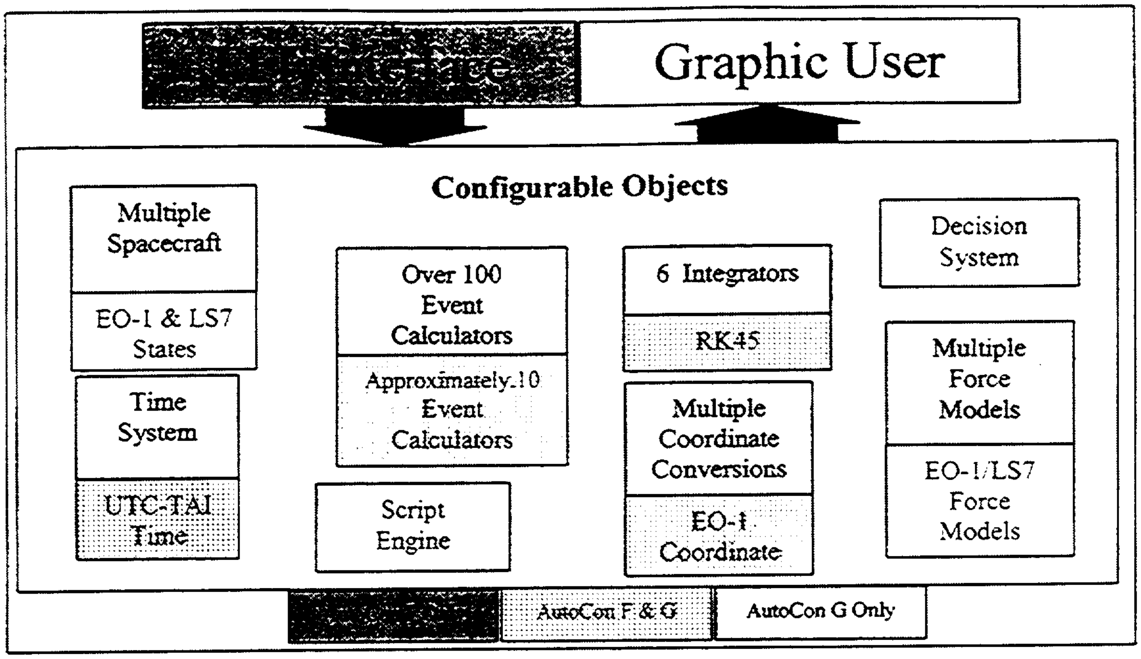
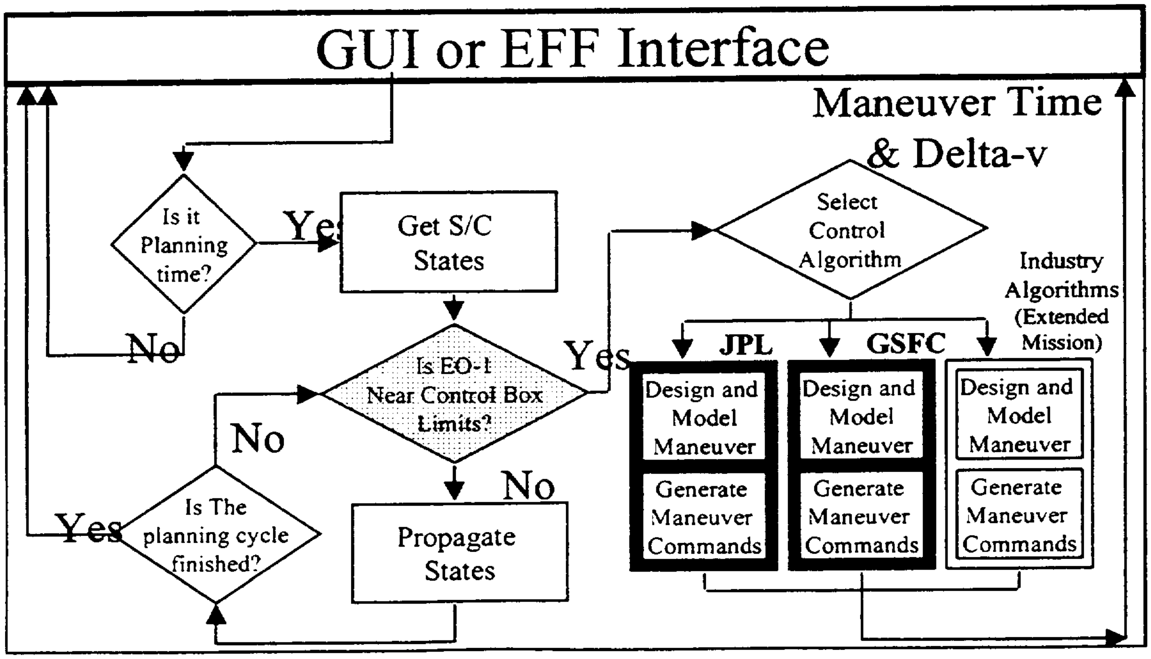


Figure 1 – AutoCon™ Events and Objects

including GPS position information, and to create command loads for computed burn times and durations. Only the objects and methods needed to support EO-1 formation flying are incorporated in the AutoCon™ -F system conserving on-board resources.
AutoCon™-F inherits from its ground-based counterpart its object-oriented C++ design. The ground system was developed with the user interface separate from the basic computational engine to provide portability and flexibility to use as flight software with minimal modifications. Scaling the existing ground software for on-board use not only saves money in porting, but also saves in testing, since the development path automatically provides a ground reference system.



GUI or EFF Interface
Maneuver Time & Delta-v
Is it Planning time?
Yes
No
Get S/C States
Is EO-1 Near Control Box Limits?
Yes
No
No
Propagate States
Is The planning cycle finished?
Yes
Select Control Algorithm
JPL
GSFC
Industry Algorithms (Extended Mission)
Design and Model Maneuver
Generate Maneuver Commands
Design and Model Maneuver
Generate Maneuver Commands
Design and Model Maneuver
Generate Maneuver Commands

Figure 2 - AutoCon Control Flow

AutoCon™ provides the architecture to facilitate interchangeable formation flying algorithms. Designed to be flexible and extendable, AutoCon™ is built around a structure called an *event*. Events can be added to AutoCon™ as necessary to support new algorithms or capabilities, thus providing extensibility. To be flexible, AutoCon™ uses natural language scripting. The scripting provides the flow control for AutoCon™. Figure 2 shows the flow control for the EO-1 mission. A new algorithm can be defined by events that are scripted to represent the algorithmic process. As long as all the necessary events exist, a new algorithm can be uploaded and executed on-board without changing the flight software.

AutoCon™ is the enabling technology for EO-1's formation flying control. AutoCon™ provides the architecture for constraint resolution, maneuver planning and handling multiple formation flying algorithms and scenarios. The architecture not only supports the GSFC and Jet Propulsion Laboratory (JPL) targeting algorithms, but provides for additional approaches as well.

EO-1 Mission requirements

NASA created the NMP to develop and validate the advanced technologies necessary to support space exploration in the 21st Century. NMP's first earth observing mission is EO-1. EO-1 has as a principal mission requirement to successfully complete 100 to 200 paired scene observations with Landsat-7 in order to validate the technologically advanced imagers on EO-1 (ref. 1). To enable the paired scene process, the EO-1 spacecraft must fly over the current groundtrack of Landsat-7 within +/-3 km. Also, in order to maintain a safety criterion, the nominal along-track separation will be one- minute +/- six seconds. The six-second tolerance is derived from the +/- 3-km groundtrack requirement. Maintaining this separation requirement is referred to as formation flying.

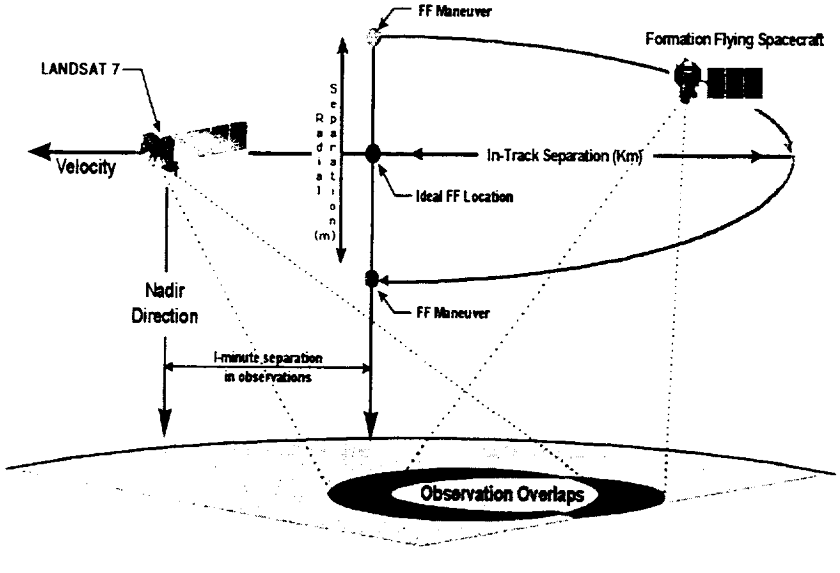


Figure 3 - Formation Control

Formation flying involves position maintenance of multiple spacecraft relative to measured separation errors. Differences in the ballistic coefficients between the spacecraft increase the relative motion and the separation errors. Therefore, maintenance of the formation may require the use of an active control scheme to maintain the relative positions of the spacecraft. Optimally, this process will be performed autonomously on-board the spacecraft and is called Enhanced Formation Flying (EFF). An example of the orbit dynamics of formation flying is shown in Figure 3.

The goal of the EO-1 EFF experiment is to validate technologies for autonomous coordinated or formation flight with respect to Landsat-7. These two spacecraft have significant differences in their ballistic coefficients. Algorithms were developed to meet the formation flying requirements. These algorithms provide EO-1 with the ability to adjust its orbit to maintain formation with Landsat-7. Initially, two algorithms will fly on EO-1: a GSFC-developed algorithm known as the Folta-Quinn algorithm (ref. 2), and a JPL-developed algorithm (ref. 3). AutoCon™ is designed to accommodate additional algorithms, as they become available.

AUTOCON™ -F

The implementation of an autonomous orbit control system facilitates lights-out operation, reducing costs and streamlining the spacecraft operations. While the motivation for implementing such a system is apparent, the challenge in this case is deriving the flight system from a ground system. Since AutoCon™ was originally designed as a ground system automation tool, a number of steps needed to be taken to convert the system for use on-board the EO-1 satellite. The steps range from changing the interface to the system, to porting the system, to scaling the size and controlling the CPU processing. The steps in the conversion were straightforward to identify. Issues raised while implementing these steps are discussed below.

Changing the Interface

The first step taken in making AutoCon™ flight-ready was to change the interface of AutoCon™. AutoCon™ was designed with a complete graphical user interface to allow the user to change inputs through dialog windows and to view results graphically. This part of the user interface would not be required for the flight version. For user friendliness, the underlying inputs to the core AutoCon™ system are ASCII files, with only a few exceptions. The flight system interface requires binary table inputs, control of the system through commands and operating modes, and the collection of results through telemetry.

The ground version of AutoCon™ was first modified to accept binary table files as an alternative input method to the ASCII file input. Binary tables are used because they are more compact and provide an accepted format for upload to the satellite. To accommodate all the inputs and maintain the flexibility of AutoCon™, 12 different inputs had to be converted to tables. Because EO-1 has a table size limit of 3000 bytes, the planetary input data had to be broken out in to three separate tables.

To ensure table data integrity, AutoCon™ was implemented with the capability to validate the tables before use. Validation design required that all tables include three fields. The first two fields in the table are the table identifier number and the table size in bytes. These two fields are checked to ensure that the correct table was uploaded and accessed by AutoCon™.

The last field in all tables is a checksum that is computed using a standard 32-bit Cyclic-Redundancy Check (CRC) method.

Implementation of this interface upgrade provides AutoCon™-G the capability of ingesting data in either the table format or the ASCII format for each input. This approach allows for systematically testing each input table, and provides a complete code base with which to begin porting to the flight system environment.

Porting from PC/Windows NT to the Mongoose V/VxWorks

AutoCon™-G was developed under Windows NT. The flight system is built around a Mongoose V (MGV) processor with a VxWorks operating system. The system is built with the Tornado compiler, which is a derivative of GNU for this environment. Since a MGV system was unavailable at the time of the initial port, the approach was taken to first port AutoCon™ to a similar system. AutoCon™ was ported to HP UNIX and built with the GNU compiler. Although AutoCon™ was designed from the beginning to be portable, this step was the first real test of the system portability. The following issues were addressed during this port.

The first issues were the easiest to resolve technically. Because the UNIX environment is case sensitive for filenames, all references to included files had to reflect the true file name. As a simple resolution, all filenames and references to filenames were changed to lower case. Next, the system had to be built in the environment. The AutoCon™ system was originally designed as a collection of dynamic link libraries (dll's) with an executable driver for the Windows environment. For the flight system, a single executable had to be built. This change was resolved by generating an all-inclusive makefile. The major porting issues came during the compiling and linking of the system.

The compilation issues that had the largest scope were the use of the GNU string and math libraries. To resolve issues with the compiler provided string class, a simplified string class specific to AutoCon™ was developed to override the system string class. To support the change in the math library, AutoCon™ required its own definition of PI and the re-evaluation of error handling for the math functions. For example, the fmod function provided in the Windows environment returned a value of zero when one of the two passed arguments was zero. In the flight environment, a +NAN (not a number) was returned from the same method if a zero was passed as the second argument.

Other compilation issues were associated with compiler limitations from the ANSI standard. One such limitation was the return of a value of an indexed array as demonstrated in Figure 4.

Original Code

```
return array_value[index];
```

Fixed Code

```
float temp_value = array_value[index];
return temp_value;
```

Figure 4 – Resolution to Compiler Limitation

The largest obstacle to porting to the flight system was the restriction on dynamic memory allocation. While the object-oriented design of AutoCon™ is based on the creation of objects at run time, the flight operating system forbade the use of dynamic memory allocation. To overcome this hurdle, AutoCon™-F was fitted with a memory manager. The memory manager contained in its data segment a 1,500,000 byte block of space, and overrode the C++ new and delete operators to manage the use of the space for AutoCon™. A number of redesigns were implemented before the memory manager operators would compile without conflicts with other parts of the flight system.

Size Reduction

The next challenge was to fit the AutoCon™ system into the available space on-board the satellite. Since AutoCon™ was originally designed as a ground system, size of the system was not a major concern. Although AutoCon™ is a relatively small Windows system, it exceeded the size limitations for flight code. When first ported to the UNIX environment as a single executable, the AutoCon™ executable size was over 7 Mb. The spacecraft requirement was to get AutoCon™ under 500 kb in the flight environment.

Figure 5 shows the AutoCon™-F size history after the first build in the Mongoose V environment, which includes an initial attempt at size reduction. The first five months include seven builds that were focused solely on reducing the size of the system. Once the size requirement was achieved, subsequent builds focused on modifying the capabilities of AutoCon™-F to support the flight system interfaces and mission requirements. As new capabilities were added, the code size was re-evaluated and reduction efforts were implemented.

The first step was to remove unnecessary capabilities. Since AutoCon™ is object-oriented this simply entailed removing whole objects from the build of the system. The original system contained 256 classes that defined the objects. The first two builds removed a total of almost 160 classes. Most of these classes are related to the calculation of event data that was not necessary for flight requirements. To remove additional objects, AutoCon™ was modified to use the math functions of the Attitude Control System (ACS) in place of its own math classes. The final core AutoCon™-F system includes 85 C++ classes.

While removing the objects offered a significant reduction, the system size still exceeded the spacecraft requirement. The next step in code reduction was to eliminate code methods from within the remaining classes. Methods associated with file input and debug output were obvious candidates for removal, as were methods required by these. Methods associated with unnecessary coordinate transformations and flight regimes outside of the EO-1 orbit were also omitted.

Another size reduction technique explored was compiler flag settings. Figure 5 shows a large size reduction in February 1998 between the 6th and 7th builds. The primary difference between these two builds was that the flag for compiling with debug was turned off for the 7th build; it was activated for all previous builds.

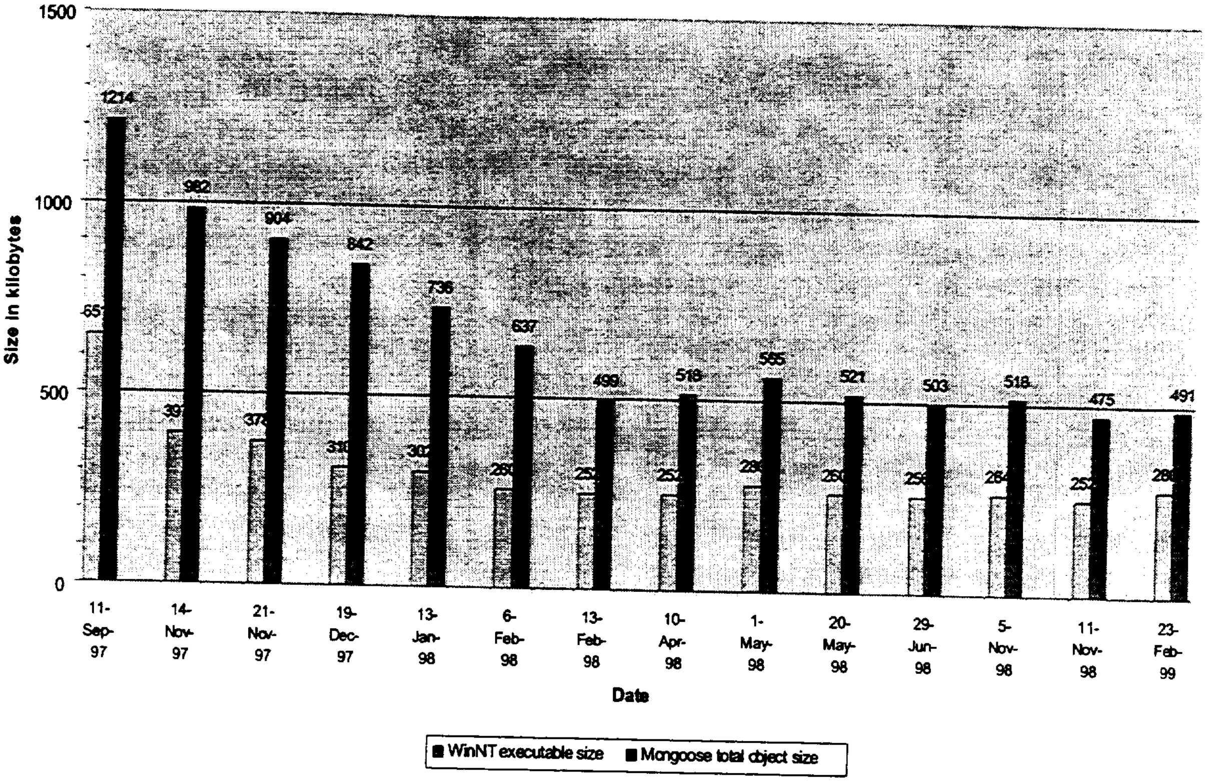


Figure 5: AutoCon™-F Size History

During subsequent size reduction activities, other methods were implemented to reduce the size of the system. One technique used was to collapse the inheritance of derived classes into their base class. This technique proved to provide minimal savings at approximately one kilobyte per collapse. Another technique used, which provided significant savings, was the static allocation and initialization of arrays. Initialization methods in some classes were filling large arrays using direct element-by-element assignments. Two classes had a combined total of 943 elements initialized by direct assignment. The change in initialization saved over 36 kb of space.

Parsed CPU Execution

The EO-1 flight environment system requires that the executing tasks use CPU time slices. The AutoCon™-F system receives a CPU slice every two seconds. AutoCon™-F is expected to complete processing within a fraction of a second. If AutoCon™-F, or any task, fails to complete processing within 5 seconds, the flight system will perform a warm restart.

AutoCon™ was originally designed as a parsed execution system for UI messaging such that one script command executes and processing returns to the controlling UI system. AutoCon™-F simply retained this design for the interface with the flight system. AutoCon™-F operates through the sequential execution of scripted commands. Using the original design, AutoCon™-F executed one command per time slice. The commands however took varying amounts of time to execute. Some commands used little CPU processing while others exceeded the allotted time slice. The commands that exceeded the time slice were segmented to complete a portion of the command, return processing to the main system, and continue with the next part of the command segment at the next time slice. To make the best use of the available time slice, the commands that used very little processing time were identified and grouped with the next command within the same time slice.

Near the final stages of development of AutoCon™-F, a new capability was required to provide burn durations and burn start times on the whole second. The modification was minor, requiring only a few lines of additional code. As a result of the change, AutoCon™-F consistently used too much CPU time to perform targeting. The cause of the CPU over-utilization was traced to the customer supplied fmod math function. The purpose of the fmod function was to return the remainder of one number divided by another. The function was implemented to perform successive subtractions and evaluations until the desired result was achieved. Because the use of fmod in the targeter had a very large number being divided by a relatively small number, the number of execution cycles to complete the processing was unreasonable. To ensure proper CPU use, the fmod function was replaced in AutoCon™-F with a less CPU intensive algorithm.

Issues During Testing

Upon successful compilation of the AutoCon™-F system in the flight environment, testing began. The test approach was to define a series of benchmark tests using AutoCon™-F in the Windows NT environment. The same tests were then duplicated in the flight environment and the results were compared. The flight environment tests were designed to exercise data table uplink, telemetry downlink, commanding, as well as computational accuracy. Initially, the results between the Windows NT and the Mongoose V were unacceptably different.

AutoCon™-F is required to propagate two spacecraft states (EO-1 and Landsat-7) into the future and plan any maneuvers required during that time to maintain the formation. The propagation differences between the benchmark NT result and the flight environment result were 540 meters RSS after 36 hours of propagation. The tools for debugging this problem on the flight environment were limited.

Original Code

```
if((*istat = jaccwf(a1_time, new_file, &geo)) != 0)
{
   return 0.0;
}
```

Fixed Code

```
*istat = jaccwf(a1_time, new_file, &geo);
taskDelay(1);
if(*istat != 0)
{
   return 0.0;
}
```

Figure 6: Fix to Compiler Problem

The investigation of this difference required embedding debug statements throughout the code. Since the largest perturbation on the spacecraft, besides the mass of the Earth, is drag, the drag model was investigated first. This selection turned out to be the right place to look. Something in the code was causing the model to return an atmospheric density of zero without returning a processing error. The problem was found to be in a conditional statement in the Jacchia-Roberts drag model class, where a variable was set to the result of a function call and then tested. While the code complied with ANSI standards, the compiler did not handle the syntax properly. The problem continued even when the variable assignment was removed from the conditional. The problem was fixed when an interceding function call (*taskDelay*) was added. The original code and modified code syntax is provided in Figure 6.

Original Code

```
ABOOL ASolarSystem::GetSunMoonPosition(FSIZE * s)
{
 if (fabs (s[0] - mjt) > timeTolerance ) {          // recalculate
  sunCalculated = moonCalculated = FALSE;    // must recalculate
  mjt    = s[0];                                     // calculate new times
 }
 if (sunCalculated == FALSE)  {                     // sun not yet calculated
  sun[0] = mjt;
  moon[0] = mjt;
  if ( planetaryTable.CalculateSunMoonVectors(sun,moon) == FALSE )
      return FALSE;
 }
 memcpy(&s[1],&sun[1] ,sizeof(FSIZE)*3);      // copy sun
 sunCalculated = moonCalculated = TRUE;
 return TRUE;
}
```

Fixed Code

```
ABOOL ASolarSystem::GetSunMoonPosition(FSIZE * s)
{
 if (fabs (s[0] - mjt) > timeTolerance ) {          // recalculate
  sunCalculated = FALSE;
  moonCalculated = FALSE;                          // must recalculate
  mjt    = s[0];                                     // calculate new times
 }
 if (sunCalculated == FALSE)  {                     // sun not yet calculated
  sun[0] = mjt;
  moon[0] = mjt;
  if ( planetaryTable.CalculateSunMoonVectors(sun,moon) == FALSE )
      return FALSE;
 }
 memcpy(&s[1],&sun[1] ,sizeof(FSIZE)*3);    // copy sun
 sunCalculated = TRUE;
 moonCalculated = TRUE;
 return TRUE;
}
```

Figure 7 - Fix to Another Compiler Problem

After the problem with the atmospheric drag modeling was found, there was still approximately 36 meters of propagation difference using the full force model. To investigate this difference, separate results were produced for each force in the force model. This force-by-force testing showed that all forces modeled except for geopotential produced small (sub-meter) errors, but the largest discrepancy was related to the effects of the moon. Further tests revealed that, on the Mongoose, after the initial calculation of the moon's position the moon's position remained static, while the epoch was being advanced.

Inspection of the code shown in Figure 7 revealed the same type of structure found in the drag problem: the calculation and testing of a value in a conditional statement. This statement was broken apart, but did not correct the problem. Further analysis revealed that the error was being caused by a chained assignment. The correct path was being executed, but the variables were not properly updated, causing the subsequent test to incorrectly bypass re-calculating the position of the sun and moon. The correction required breaking the chained assignments into separate statements. Chained assignments were subsequently broken up in all other parts of the code even though they were not currently experiencing problems.

Once the compiler issues were resolved, the comparisons between the Windows NT and the Mongoose V results agreed.

GPS DATA SMOOTHER

Introduction

Ensuring an accurate input state to AutoCon™-F is crucial for lights-out operation. On EO-1, the GPS TENSOR™ software using a Kalman filter processes raw GPS data that consists of pseudorange and Doppler measurements. Orbital states obtained from the GPS TENSOR™ have RMS position and velocity errors of 35.7 m and 5.2 cm/s, respectively (ref. 4). The requirement for an AutoCon™-F input state for the GSFC algorithm is that the errors in radial position and velocity be no larger than 5 m and 2 cm/s, respectively. Thus, the GPS TENSOR™ solution alone is not adequate, and an additional stage of optimization must be provided. This stage has been implemented as a discrete fixed interval data smoother, which uses the Rauch, Tung and Striebel (hereafter, RTS) algorithm (ref. 5). While the solutions provided by the GPS TENSOR™ software are actually processed, they will be referred to in this section as *measurements*. These measurements are assumed to be from a converged Kalman filter solution.

The RTS algorithm is, itself, based on an iterative Kalman filter. Each of N measurements is collected and processed, while storing at each interval the filter's a-priori state and state error covariance matrix (current iterate's state and error covariance advanced to the current measurement epoch), the state transition matrix, and the optimized (a-posteriori) state and error covariance. These stored quantities are referred to collectively as the *data arc*. Measurements are processed at one-minute intervals, and a number of measurements, corresponding to somewhat more than one orbital period (N~120), are required for optimal state determination. The RTS algorithm is a backward iteration through the data arc, taking the initial smoothed state to be the final filter computation in the forward sweep. In this way, the original filter estimate is updated to provide an improved smoothed estimate based on all the measurement data collected. Unlike the situation with the Kalman filter (forward sweep) updates, the smoothed state error covariance is not required in the computation of the smoother estimates in the backward iteration.

While, in principle, the RTS algorithm allows for computation of smoothed estimates at any or all of the interior intervals, care must be taken to ensure that the estimate is meaningful. Thus, the following convergence criteria have been established for the smoothed estimate, and the backward sweep is terminated if one of them occurs:

- a diagonal element of a current iterate's error covariance matrix is not positive definite
- position or velocity error diverges; i.e., a current iterate's RSS position or velocity error exceeds that of the previous iterate.

The first criterion is required because of the decoupling of the error covariance computation from the state estimate. The second criterion simply ensures that the state estimate converges in the usual sense. Two additional convergence criteria may be turned on in operation, causing the backward iteration to terminate when:

- a current iterate's RSS position or velocity error exceeds a commandable tolerance
- a current iterate's state element estimate is out of bounds.

Kalman Filter

The Kalman filter underlying the smoother is adapted from the GPS Enhanced Orbit Determination Experiment (GEODE)-lite software (ref. 6). The state elements consist of the three components of position and velocity, together with a drag term coefficient, and the GPS receiver timing bias and bias rate. The dynamic quantities are computed in Mean of J2000 coordinates. The measurement model assumes that the point solution measurements are converted to Earth-Centered-Earth-Fixed (ECEF) coordinates. This model has been enhanced to incorporate the velocity components that are also provided by the GPS TENSOR™ software, and to incorporate the Jacchia-Roberts drag model, required by AutoCon™-F.

The Kalman optimization is performed in two stages: (1) propagation of the current estimate to the new measurement epoch, and (2) updating of the a-priori state due to the new measurement data. The original GEODElite code was modified so that AutoCon™-F performs the propagation step. AutoCon™-F time conversions are also called by the modified GEODElite code.

Smoother Integration and Testing

The GPS data smoother is implemented as an AutoCon™-F object, and appears to the user just as any other object does. It is created with an associated spacecraft object that holds the final smoothed state. This smoothed state is the input state for maneuver planning. As with other phases of the flight code conversion, the smoother used a parsed execution scheme, i.e., the two stages of Kalman optimization are handled in separate execution cycles. In the back sweep, only a small number of iterations are performed in each cycle to prevent CPU task overloading.

As with other facets of AutoCon™-F, the smoother is table-driven, and may be monitored via telemetry packets. Tables exist to control the operational modes of the smoother, the process and measurement noise characteristics, the data arc characteristics (e.g. queue size), as well as to provide data required for performing coordinate transformations. New data may be uploaded to the spacecraft via these tables in order to alter the secular behavior of the smoother. The uploaded tables are validated and checked for integrity in the same way as for the other AutoCon™-F tables. Similarly, the smoother measurement acquisition cycle and final state may be monitored dynamically through telemetry packets.

In testing scenarios, the definitive smoother state has proven to be nearly always better in comparison to a reference ephemeris than the Kalman estimate. Indeed, despite the lag in time between the Kalman and smoothed estimates, multi-day propagation of the smoothed solutions are comparable to, and often much better than, the propagated Kalman estimates, producing the desired behavior. This result may be related to the fact that the definitive angular momentum and specific energy are better determined for the smoothed states than for the Kalman filtered states.

Operational Modes

The smoother is designed to provide a state estimate with minimal ground support under normal operating conditions. The initial seed state is derived from the most recently updated GPS TENSOR™ data. This can be changed, however, by uploading the seed state and error covariance to the EFF. Usually, it is desirable to allow the smoother to complete the back sweep until one of the convergence criteria is encountered, as this process provides the best definitive solution. However, the smoother may be commanded to provide a solution at a fixed amount of time (i.e., *lag interval*) before the final measurement epoch. Setting the lag interval to 0, for example, will tell the smoother to provide only the iterated Kalman filter estimate. Indeed, the smoother may even be commanded to provide the GPS TENSOR™ state alone.

SAFETY

One of the major concerns of the EO-1 mission is to make sure that the autonomous maneuver system is as safe as possible. There was considerable concern, for example, that an autonomous system would cause the thrusters to fire for too long and jeopardize the mission. Several safeguards were created to alleviate such concerns. These include a standard of 48 hours notice before any planned maneuver (the time length is adjustable) and a phased approach to autonomy. The 48-hour notice gives the ground time to review the planned maneuver before its execution. Figure 8 displays the "levels' of autonomy or phases and transition flow. These include a monitor mode, which allows burn plans to be generated and reviewed, a manual mode, which allows maneuvers to be predicted but not implemented and a semi-autonomous mode, which allows burns to be verified by the ground before execution. Even the autonomous mode can be interrupted by ground command. Also, the autonomous mode is limited to a specified number of burns before it automatically transitions back to manual mode. A complete description of the safety modes is provided in Table 1.

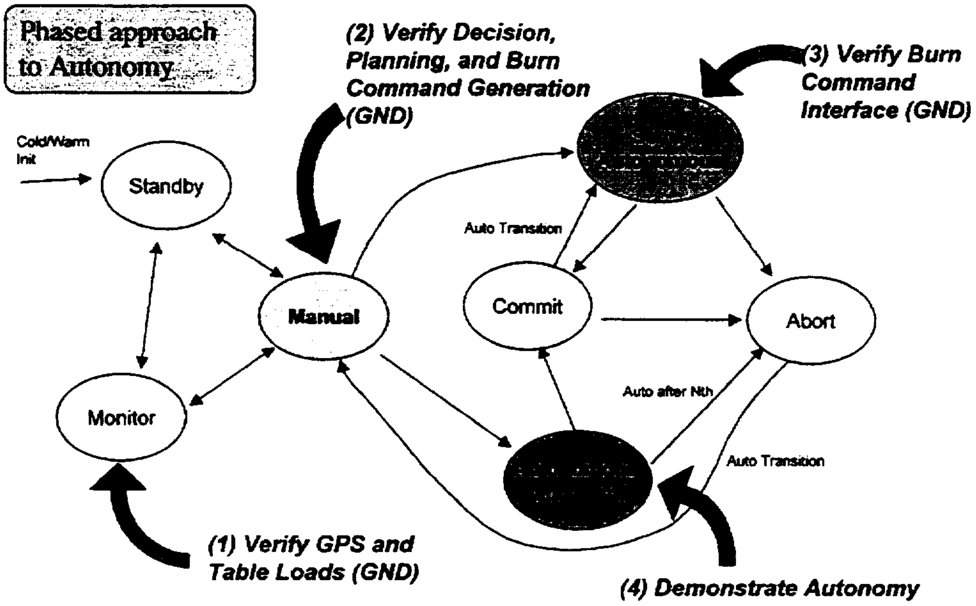

Phased approach to Autonomy
(2) Verify Decision, Planning, and Burn Command Generation (GND)
(3) Verify Burn Command Interface (GND)
Cold/Warm Init
Standby
Auto Transition
Manual
Commit
Abort
Monitor
Auto after Nth
Auto Transition
(1) Verify GPS and Table Loads (GND)
(4) Demonstrate Autonomy

Figure 8 - Control Safety Modes

Standby
- Pend on incoming data and send it to the bit bucket
- Otherwise do nothing

Monitor - (AutoCon-F executes with maximum safety for S/C)
- Invoke AutoCon-F only
- Report maneuver planning data to ground
- No maneuver commands are generated

Manual - (AutoCon-F executes with ground as safety)
- Generate maneuver commands (table loads) and send to ground only
- All burns must be command from the ground in their entirety
- Ground can loopback command from EFF telemetry if desired to execute burn

Semi-Autonomous - (Ground still in loop for go/no-go)
- Send maneuver commands (table loads) to the Stored Command Processor (SCP)
- Do not enable Absolute Time Sequence (ATS), Relative Time Sequence (RTS) in SCP
- Must switch to Commit Mode to allow loaded burn to execute
- Inaction will cause loaded burn to expire

Commit - (allow an EFF loaded burn to execute)
- Enable ATS and RTSs in SCP to permit loaded burn to be executed
- Required at least 2 hours before time of burn
- Autonomously switch to Semi Autonomous Mode upon completion

Abort - (abort an EFF loaded burn and clean up)
- Disable the ATS and RTSs in SCP to prevent execution of burn
- Clean up from any preparation for burn
- Autonomously switch to Manual Mode upon completion

Autonomous - (allow EFF to control the orbit)
- Closed loop orbit maintenance
- Use Commit Mode to switch back to Semi-Autonomous Mode and not abort a planned burn
- Ground can still monitor with 48 hour notice to burn
- Switch to Semi-Autonomous Mode after N burns. Safety for unattended operation

Table 1 – Safety Modes

In addition to AutoCon™-F's built-in safety features, the attitude control system (ACS) limits all burns to 60 seconds or less. The stored command sequence also limits burn duration. Additionally, EO-1 has a watch dog timer to make sure no task, such as AutoCon™-F, exceeds CPU utilization, depriving other critical tasks processing time. Finally, the spacecraft has a safehold mode that can disable AutoCon™, if necessary.

Timeline/Operations

The post-launch phase of the EO-1 mission is divided into three major periods. For 30 to 90 days just after launch, the spacecraft will enter a check out phase. During this time, AutoCon™-F will be turned on to run in the monitor mode. This period will begin the validation process for the GSFC algorithm. The AutoCon™-F generated burn plans will be compared with expected burn plans and ground operations.

Following the checkout phase, the mission phase will begin. At this point AutoCon™ will enter manual, semi-autonomous and, finally, autonomous mode for the GSFC algorithm, and during each step, the algorithm's performance will be validated. After about 3 months of operation under the GSFC algorithm, the JPL algorithm will be uploaded to the spacecraft and the validation process repeated at each step to validate the JPL algorithm.

Finally, during the extended mission, the industry provided algorithms will be uploaded to the spacecraft and tested in the same manner. Currently two industry algorithms are planned; however, because of AutoCon™-F flexibility, at least one more approach is under consideration and even more are possible.

SUMMARY

The push for low mission cost and multiple spacecraft support has made satellite orbit control autonomy a priority for future missions. While ground autonomy has progressed, true savings, both in cost and error prevention, require onboard autonomy. In addition, new mission concepts that require spacecraft formations demand onboard maneuver capabilities. Without formation control, many future science missions would not be possible. The AutoCon™ conversion of a ground-based mission planning and support system to a flight system shows that technologies such as autonomous control can be implemented. The EO-1 mission will demonstrate in a monitored phased approach that autonomous formation control can be realized.

REFERENCES

1. *Mission Requirements Request (MRR) for the Earth Orbiter-1 (EO-1)*, 501-NMP/EO-1 MRR01, NASA GSFC Code 501, January 1997.
2. Folta, D., and Quinn, D., "A 3-D Method for Autonomously Controlling Multiple Spacecraft Orbits", *IEEE 1998 Aerospace Conference Proceedings*, 98TH8339, March 1998.
3. Guinn, J., "Autonomous Navigation for the New Millenium Program EO-1 Mission", *AIAA Guidance Navigation and Control Conference*, AIAA 97-3816, August 1997.

4. Quinn, D. "Preliminary GPS Acceptance Test Results" April 2, 1998, NASA GSFC Code 572, Greenbelt, Maryland
5. Rauch, H.E., Tung, F., & Striebel, C.T. [RTS], *AIAA J.*, vol. 3. no. 8, 1965
6. "Global Positioning System (GPS) Enhanced Orbit Determination Experiment (GEODE) Mathematical Specifications", Revision 1 March 1997, NASA GSFC Flight Dynamics Division/Code 550, Greenbelt, Maryland

THE INTERNATIONAL SPACE STATION INTERIM CONTROL MODULE ATTITUDE DETERMINATION AND CONTROL SYSTEM– A CASE STUDY

Walter D. Grossman
Stephen Fujikawa
Applied Systems Engineering, Inc.
wgrossman@ase.wl.com

ABSTRACT

The design of the attitude determination and control system for the International Space Station Interim Control Module (ICM) presented challenges not encountered with conventional spacecraft.

The project had a very short schedule so the primary challenge was to develop a *simple* controller which performed well for the multitude of configurations and construction phases for which mass properties vary by orders of magnitude. The vehicle had to perform as a free-flyer with the center-of-mass (CM) located at the system geometric center. The vehicle had to support the "L"-shaped 4A.1 configuration for which the CM is not even located on the spacecraft body. As a free-flyer there was minimal gravity gradient. In 4A.1 gravity gradient torques can exceed 60 newton-meters. As a free-flyer, the system input gain is $\tau/J \approx 2 \times 10^{-2}$ radians/sec^2. In the larger configuration 4A.1 the system input gain is $\tau/J \approx 8.5 \times 10^{-5}$ radians/sec^2.

The attitude determination system for the ICM used a fiber optic Inertial Measurement Unit (IMU) and two wide-angle Field-Of-View (FOV) star cameras providing quaternion updates every 10 seconds. Due to concerns regarding the IMU bias stability, a six-state variable gain Kalman filter was implemented. To minimize the numerical calculations for the propagation of the six-by-six covariance equation, a novel reduced-order covariance equation was implemented which exploits the property that the full-attitude quaternion measurement yields nearly spherical attitude and bias covariance submatrices.

Reliability concerns led to the implementation of a four computer voting scheme, two independently controlled thruster manifolds, two IMUs, and two implementations of the Kalman filter. Managing this wrought redundancy was very complicated. Synchronization of the computers required insertion of computation delays which reduced significantly the system phase margin. The significance of this reduced phase margin was demonstrated in flex studies in which the simulated elimination of this delay completely eliminated the excitation of a flexible mode.

On later missions the ICM is required to provide momentum desaturation of the Control Moment Gyro (CMG). A fuel-optimal solution using the SIMPLEX method is implemented in real-time. Furthermore, the thruster system had be able to "take-over" the control with fully loaded CMGs in the event that the ISS-ICM interface became disrupted during desaturation.

This paper shall explore the design problems encountered and describe and critique the solutions used to overcome the problems.

MISSION REQUIREMENTS

The Interim Control Module (ICM) services the International Space Station (ISS) by providing attitude determination, attitude control, and orbit-raising delta-V propulsion. In it latter stages, the ICM performs the momentum management of the ISS Control Moment Gyros (CMG).

The ISS shall be built and flown at a shuttle orbit of approximately 150 nautical miles, 51 degree inclination. Depending upon the beta angle, the ISS shall be flown in one of three configurations:

1. X-Velocity Vector (XVV).
2. XVV with Torque Equilibrium Attitude (TEA) offset.

3. X-Perpendicular to Orbit Plane (XPOP).

For shuttle orbiter docking, the ISS shall also be capable of flying at arbitrary attitudes relative to the Local Vertical (LV) reference frame. The ISS shall also be capable of flying at arbitrary inertially fixed attitudes. The ICM control requirement is to maintain the ISS to within ±3° of the desired attitude during most of the mission. During orbiter approach and docking operations, the ICM must control the ISS to within ±0.8° of the desired attitude. The ICM shall control the rates of the ISS. During slews, the ICM is required to slew the ISS at no less than 0.1°/second but not allow rates to exceed 0.2°/second. The ICM attitude determination system is required to provide attitude knowledge accurate to ±0.5° per axis.

HARDWARE DESCRIPTION

Actuators

The ICM provides attitude control with a bipropellant Reaction jet Control System (RCS). This system consists of twelve 22 N (5 lbf) thrusters and two 110 N (25 lbf) thrusters. The delta-V main engine is a fixed to the ICM and provides 484 N (110 lbf) of thrust. The ICM has no reaction wheels nor other momentum exchange devices.

The ACS thrusters are arranged in two manifolds to provide a measure of fault tolerance. Both manifolds are used in normal operations. The system remains fully functional in all modes in event of the failure of either manifold, although the available control authority is reduced. To avoid contamination effects, forward facing thrusters are nominally not used, although they may be enabled to increase available torque in the event of a manifold failure. Figure 1 illustrates the layout of the ICM RCS.

Because of the large variation in the Center-of-Mass (CM) location over the different construction phases, the force vector of the main engine imparts significant attitude disturbance torques during the delta-V maneuvers.

Sensors

The ICM provides attitude determination data to the ISS using two STar Cameras (STC) and three Inertial Measurement Units (IMUs). The STCs output a star field to spacecraft computer from which an attitude quaternion is derived using an algorithm supplied by the STC manufacturer. Due to processing constraints, only one STC is used per 10 second sensor cycle to update the attitude estimate.

Two fiber optic IMUs operate simultaneously and the third is carried as a cold spare. Each IMU propagates the dynamics of an associated Kalman filter (KF), with each KF maintaining the estimate of the bias associated with its IMU. The IMUs operate output three delta-theta rotations at 100 Hz. These samples are accumulated in a 20 sample FIFO to yield vehicle rotation measurements at 5 Hz.

System Mass Properties

The ICM is designed to support the ISS from construction phase 2A.1 through phase 7A (and possibly further). The mass properties of the combined ISS/ICM vary greatly throughout construction. Unlike more common space vehicles, the CM is often not located within the physical body of the spacecraft. The inertia tensor varies greatly during construction and there are significant product of inertia terms. A sudden change in mass properties occurs when the shuttle orbiter docks with the ISS. The mass properties are summarized in Tables 1 and 2.

Computer Hardware

The ICM processing hardware consists of four computers. The four computers operate independently, with each one receiving the full complement of sensor information and returning a bit-mapped

Table 1: Center of Mass Location (meters), ISS Analytical Coordinate System.

Mission Phase	CM_x	CM_y	CM_z
ICM (only)	-24.99	-0.076	4.14
2A.1	-14.81	0.02	4.94
3A.1	-13.09	0.00	4.40
4A.1	-11.30	0.00	0.85
5A.1	-9.04	0.03	1.62
6A.1	-8.59	-0.05	2.28
7A.1	-12.85	0.24	2.68
2A.1+shuttle	-3.51	0.02	-3.28
3A.1+shuttle	-3.86	0.01	-2.77
4A.1+shuttle	-0.37	0.01	6.84
5A.1+shuttle	-0.11	0.02	6.78
6A.1+shuttle	1.52	-0.04	9.90
7A.1+shuttle	1.07	0.13	9.56

thruster "fire word" by which thrusters are turned on and off. The computers are configured into two pairs of two strings. The output fire words of the two computers in a string are bit-wise "AND"ed together. The two resulting fire words from each string are bit-wise "OR"ed to yield the final output fire words.

ATTITUDE DETERMINATION AND CONTROL

Nominal Control

The ICM control system is an all-gas reaction jet system. Several control methods were examined including various forms of Pulse-Width Modulation (PWM) and the bang-bang phase plane deadband controller. PWM controllers demonstrate finer control capability than deadband controllers and they are amenable to classical linear analysis. The PWM controller caused more thruster cycles than the deadband controller. The PWM is more complicated to implement, requiring a separate 100 Hz software layer (the control loop operates at 5 Hz) to generate the 10 millisecond pulses. The deadband controller provided control capability adequate to meet the control system requirements. Its implementation was simpler than the PWM, requiring a fire/don't fire decision on every 5Hz control loop cycle. The deadband system was chosen primarily for its adequacy and simplicity.

The deadband controller caused many fewer thruster cycles than the PWM. The issue of thruster cycle life arose well after the design decision was made. The selection of the deadband controller, though not based upon thruster cycle life, was good one from this perspective.

The thruster switching commands are determined using dead-band switching plane for each of the three axes. The ICM uses a novel switch plane implementation. The abscissa of the switch plane is not an Euler angular error but rather the quaternion parameterization:

$$\theta = 2 \operatorname{sign} q_4^\epsilon \begin{bmatrix} q_1^\epsilon \\ q_2^\epsilon \\ q_3^\epsilon \end{bmatrix} \tag{1}$$

where $q^\epsilon = \hat{q} \otimes q^d$, where $\hat{q}$, q^d are the estimated attitude and the desired target attitude, respectively. Quaternion elements q_1, q_2, q_3, q_4, represent, respectively, the x, y, z, and scalar parts of the quaternion.

Table 2: Inertia Tensor (kg-m^2) for Different Construction Phases, ISS Analytical Coordinate System.

Mission Phase	J_{xx}	J_{yy}	J_{zz}	J_{xy}	J_{xz}	J_{yz}
ICM (only)	5.0×10^3	1.0×10^4	1.0×10^4	3.8×10^1	1.1×10^2	-2.8×10^2
2A.1	2.7×10^5	2.9×10^6	2.8×10^6	-4.5×10^3	-2.0×10^5	-1.5×10^3
3A.1	4.4×10^5	4.0×10^6	3.7×10^6	6.4×10^3	7.7×10^4	-9.5×10^3
4A.1	3.5×10^6	7.2×10^6	5.8×10^6	-2.1×10^4	1.6×10^6	-4.2×10^3
5A.1	3.9×10^6	9.7×10^6	8.1×10^6	-2.5×10^4	8.6×10^5	1.9×10^4
6A.1	4.3×10^6	1.1×10^7	8.8×10^6	3.7×10^4	8.3×10^5	3.7×10^4
7A.1	4.6×10^6	2.3×10^7	2.1×10^7	-2.3×10^5	1.8×10^6	-4.6×10^4
2A.1+shuttle	1.5×10^7	2.8×10^7	1.4×10^7	-1.6×10^3	6.5×10^6	-1.4×10^4
3A.1+shuttle	1.5×10^7	2.8×10^7	1.3×10^7	-5.2×10^3	6.4×10^6	-1.4×10^4
4A.1+shuttle	9.1×10^6	3.6×10^7	2.9×10^7	-2.2×10^4	-6.1×10^6	-1.2×10^4
5A.1+shuttle	9.2×10^6	3.6×10^7	3.0×10^7	-5.1×10^3	-6.1×10^6	2.3×10^4
6A.1+shuttle	2.4×10^7	5.0×10^7	2.9×10^7	3.2×10^3	-1.3×10^7	-2.6×10^3
7A.1+shuttle	2.4×10^7	5.0×10^7	2.9×10^7	1.9×10^5	-1.3×10^7	1.5×10^5

The parameterization of (1) is simpler than the Euler angle parameterization. The real advantage of (1) is that it is well behaved for all attitudes and its implementation has yielded a controller with demonstrated global convergence. While early concepts of the control law (into the preliminary design review) specified a separate mode for large angle slew, the global convergence of the control law using (1) obviated the need for this separate mode.

The ordinate of the switch plane is an estimate of the ICM rigid body rate. The rate of the ICM is measured by the IMU. Each sensed axis rate is passed through a six-pole filter to provide high frequency roll-off or possibly "notches" to mitigate the effect of parasitic structural resonances. A typical ICM switching curve is shown in Figure 2. One of the analytical models used for stability analysis is shown in Figure 3. Note the presence of the rate filter $H_r(z)$ in the rate feedback loop.[1]

Thruster Pulse Restrictions

The control system design and analysis was complicated by a multitude of ad hoc restrictions on thruster usage that emanated from the many groups analyzing ISS/ICM performance. The restrictions were not delineated at the program beginning, rather they evolved as certain analysis groups completed their work. This "evolution of requirements," perhaps unavoidable in projects of the magnitude of the ISS, greatly complicated the ICM ACS development and design. Typically, after a design been completed and simulated a new restriction was imposed. The effect of the restriction had to be simulated *over a span of several stressing configurations* to determine the effect upon vehicle performance. In certain situations, particularly when the vehicle suffered the loss of a reaction jet or manifold, the system failed to perform with the given restriction. In those situations the imposition of the restriction had to be "fought" and limited. One of the main goals of the ACS group was to identify those situations and to do so rapidly before the restriction became a formal requirement.

A summary of some of the restrictions are listed in Table 3.

[1] This model was used in part of the stability analysis. Note the absence of any star camera as an absolute attitude reference. The elimination of the star camera simplified the analysis while not affecting the results of the short-term stability analysis.

Table 3: Thruster restrictions imposed on the ICM ACS system.

Restriction	Reason
System should limit thruster cycling as much as possible.	To maximize thruster life.
Control Loop cycle had to be faster than 1 Hz .	To prevent thermal cycling of the jet nozzle.
No 5 lbf and adjacent 25 lbf thrusters on simultaneously.	To prevent adverse heating of the nozzles.
Certain to-be-defined pulse spacings could not be used.	Limit mechanical loads on specific structural parts.
The two forward-pointing pitch thruster should not be used.	To minimize the exposure of the ISS to debris from thruster exhaust.
No thruster shall be turned on within a 0.5 second of the shutdown of any other thruster.	Concern for potential thruster instability caused by pressure-wave induced fuel/oxidizer mixture perturbation. (The "water hammer" effect.)

Implementing the restriction on Table 3 was complicated by synchronicity issues arising from the four computer system implementation.

Control Loop Timing

In the early stages of design, both the attitude determination process and the control process loop frequency, partly based upon Naval Research Laboratory (NRL) heritage, were set at 20 Hz. This high frequency appeared unnecessarily fast for this massive vehicle so out of a desire to reduce computational throughput and reduce thruster cycling we undertook a review of the control loop frequency. From a series of simulation runs, it appeared that 1 Hz control loop frequency (yielding 1 second pulses minimum on time) was appropriate. Other development centers in the program had developed simulators that ran at 5 Hz so for the sake of conformity, the ICM control loop cycle time was also set to 5 Hz.

For reasons not forseen at the time, selection of this higher rate was fortunate. In order to synchronize the output fire words of the four computers it was necessary to insert a single loop cycle delay between the time the fire word was calculated and when it was executed. (Note unit delay in forward path, Figure 3.) This additional lag is always detrimental to a control system. In the larger configuration, the effect of this additional lag is not significant. In the smaller configuration when the ICM acts as a free flyer[2], this additional lag may cause oscillatory closed-loop behavior. As the lag increases, the system becomes unstable.

Reboost/Deboost

Reboost maneuvers are carried out on a periodic basis to maintain the orbit. The 100 lbf VJ1 main engine is used for delta-V and the ACS thrusters are used to control attitude with the same control laws as XVV flight. Because the VJ1 thrust vector is fixed, the CM offset varies with the ISS mass configuration, and different disturbance torques will be generated. The VJ1 mounting angles are designed to mitigate these effects, but they are still many times larger than worst case environmental torques. In the case of a failed manifold, reboost is still possible, but in some ISS mass configurations

[2] This configuration may occur at the initial deployment and during the decommisioning deboost (re-entry) phase of the program.

the VJ1 disturbance torque exceeds the maximum possible restoring torque provided by the ACS thrusters. In this case, VJ1 is off pulsed automatically when attitude error exceeds 5° to allow the ACS to reduce the pointing error. Hysteresis is added to the VJ1 controller such that the engine is not turned back on until the error is reduced below 2°. This hysteresis increases VJ1 duty cycle.

Deboost is a bus disposal maneuver and involves the ICM alone separated from the ISS. Delta-V is provided by the 110 N (25 lbf) jets which are off pulsed and used in conjunction with the 22 N (5 lbf) jets for attitude control. Deboost is the most difficult control mode and requires the most thruster activity and propellant usage. This increased thruster activity and fuel usage occurs because the ICM-only mass properties are many orders of magnitude smaller than the when it the ICM is mated to the ISS while the minimum impulse bit remains the unchanged for all the configurations. These effects are aggravated during deboost because the ICM fuel is nearly expended. As a result, ACS behavior during deboost is characterized by rapid two sided limit cycling. In order to obtain sufficient damping, it is necessary to increase the controller deadband width so that the minimum impulse bit cannot drive the system fully across it.

Flexible Body Issues

The control system was designed using rigid-body models for both the "design" model and the "truth" model. (The truth model is the one used in the simulation.) As the NRL/JSC flexible-body structural analysis progressed, the rigid-body truth model was supplanted with a flexible body truth model.

Much to the surprise of the ACS analysts, a 1.6 Hz parasitic roll resonance occurred which seriously degrade system performance. Figure 4 demonstrates the ISS/ICM rate profile for a large angle slew using the rigid body model. Figure 5 demonstrates the ISS/ICM rate profile for the same flew using a flexible body model and no rate filter. When this problem was discovered, the rate filter were adjusted to provide a high-frequency rolloff. Experiments with several different filters were performed. Two-pole Butterworth filters with corner frequencies of 0.04 Hz and 0.10 Hz were tried initially. Care had to be taken to assure a high enough frequency response for the system to reject the plume disturbance torque pulses that occurred during proximity operations. Subsequently, three sections of two-pole Butterworth filters with 1 Hz corner frequencies were concatenated to yield greater attenuation of the higher frequencies while producing less lag than the slower filters. In addition, the switch plane slope and deadbands were modified.

In all situations it was possible to mitigate the deleterious effects of the flexible-body interaction while maintaining adequate system performance. To further reduce lag of lowpass filters, further study is currently being performed to examine the effectiveness of applying the thruster-induced accelerations into the rate filter by modeling the torques and the system mass properties.

Momentum Management

In later stages of the ISS, attitude control is maintained using Control Moment Gyros (CMGs). Due to gravity gradient and aerodynamic secular torques, the CMGs accumulate momentum which has to be shed using the ICM RCS. To accomplish this momentum management in a fuel-optimal manner, the SIMPLEX algorithm was implemented.

The RCS-based momentum management problem is a classic linear programming problem. Letting τ be the 3×14 matrix of torques produced by each thruster, t be the 14×1 vector of nonnegative thruster burn times, and H be the three element command vector of momentum to be dumped, the optimal-fuel momentum management problem is stated succinctly:

$$\begin{aligned} &\textit{Minimize} \\ J &= Ct \\ &\textit{subject to} \\ H &= \tau t, \end{aligned}$$

where C is the 1×14 "cost" vector and element C_j is the thruster force of the jth thruster divided by its specific impulse, i_{sp_j}. The restriction that burn times $t_j \geq 0$ defines the feasible set in the linear program.

The output of the SIMPLEX algorithm is a list of three thrusters and their corresponding burn times.

Complicating the algorithm further is that the momentum can be shed in a single pulse or in a pulse train prescribed by a ground-loaded pulse train timetable. The pulse generation logic is complicated in the burns have to be evenly spread out to occur at the proper fire times and the minimum pulse is 0.200 seconds. Thruster burn counters and pulse foreshortening logic had to be implemented.

Attitude Determination

The ICM attitude determination system consists of two three-axis autonomous star cameras and two fiber optic IMUs (a third is kept as a cold spare.) The star camera and IMU are similar to those used in the Clementine program.

The star camera had a wide 25° Field-Of-View (FOV). This wide FOV was a double-edged sword both simplifying some aspects of the attitude determination design while complicating other aspects. The wide FOV provided better boresight roll accuracy than the narrow FOV cameras and it provided star fixes in relatively sparse portions of the sky. This large field FOV also provided a wide entry cone for glint, structural obscuration, and allowed for blinding by sun, earth, and moon.

To identify stars, track them, and generate an attitude quaternion, the star camera manufacturer provided an algorithm which was ported onto the spacecraft computer. Once stars were identified, the software calculated spacecraft attitude using the QUEST algorithm in ref. 1. The algorithm generated various "quality" metrics including an attitude estimation residual, a count of the number of stars tracked, and a count of the number of stellar "triangles."

The original design of the ICM attitude determination algorithm was similar to the six-state fixed-gain Kalman filter used in the Clementine program. Several problems were encountered with the Clementine algorithm which led to modification for the ICM implementation. The calculation of fixed gains for a Kalman filter are determined from the steady-state solution of the Riccati equation. In typical spacecraft applications, the gains associated with the attitude correction reach steady state quickly and have a limited dynamic range. The gains associated with the bias correction converge much more slowly and have a dynamic range spanning several orders of magnitude. Using the fixed steady-state bias gain leads to a filter which converges slowly since the steady-state covariance matrix (upon which the bias gain is based) is small and measurements are not heavily weighted. An intermediate value for the bias gain selected prior to steady-state can be used leading to faster convergence but noisier steady-state performance. Furthermore, it was noticed in Clementine that the fiber optic IMU "bias stability," i.e., the time-variation in nominally fixed bias, was larger than that experienced with conventional "iron" IMUs. When prolonged update outages occurred, the attitude estimate would wander due to angular random walk and changes in the bias level. When updates resumed, the attitude estimate and bias demonstrated a strong transient correction since the bias gain was fixed and the residual was large.

A simple two-state covariance Riccati equation was used to fix these problems. The autonomous star cameras, unlike the convention star camera, outputs a three-axis attitude estimate of nearly equal measurement uncertainty in all three axes.[3] This nearly spherical measurement covariance leads to a nearly spherical state covariance. This nearly spherical covariance allows for approximate reduction of the standard 6×6 covariance matrix to a 2×2 covariance matrix. The ICM covariance equation uses ones state variable for the attitude correction and one state variable for the bias correction. With this simplification, the Riccati equation for the propagation, gain calculation, and

[3] The ICM STCs have a 1σ standard deviation of 150 μrads cross axis, 450 μrads along boresight.

update of the filter is greatly simplified. The propagation equation, $\tilde{P} = \Phi\hat{P}\Phi' + Q$ is given:

$$\begin{aligned} P_{11} &\leftarrow P_{11} + 2P_{12}T + P_{22}T^2 + Q_{11}, \\ P_{12} &\leftarrow P_{12} + P_{22}T, \\ P_{11} &\leftarrow P_{22} + Q_{22}. \end{aligned}$$

The gain equation, $K = PC'(CPC' + W)^1$ is given:

$$\begin{aligned} \sigma_\rho^2 &= (P_{11} + W), \\ K_{\text{attitude}} &= P_{11}/\sigma_\rho^2, \\ K_{\text{bias}} &= P_{12}/\sigma_\rho^2. \end{aligned}$$

The residual gate checking of residual ρ (for outliers detection and elimination) is determined by test:

$$\rho^2/\sigma_\rho^2 < N\chi^2,$$

where $N\chi^2$ is an uploaded acceptance "basket." If the residual is within the acceptance basket then the attitude and bias corrections are calculated and applied,

$$\begin{aligned} \delta_\theta &= K_{\text{attitude}}\rho, \\ \delta_b &= K_{\text{bias}}\rho, \end{aligned}$$

and the covariance update, $\hat{P} = (I - KC)\tilde{P}$, is calculated:

$$\begin{aligned} P_{22} &\leftarrow P_{22} - K_{\text{bias}}P_{12}, \\ P_{12} &\leftarrow (1 - K_{\text{attitude}})P_{12}, \\ P_{11} &\leftarrow (1 - K_{\text{attitude}})P_{11}. \end{aligned}$$

RELIABILITY AND REDUNDANCY MANAGEMENT

Actuators

As mentioned previously, the RCS is implemented using two manifolds with the thrusters geometrically "interlaced." The ICM can meet all its baseline requirements using only one manifold. In the event of failed open thruster, the manifold can be shut down. Thruster failure is determined by the ground operators and they alone can shut down a manifold. No mechanism for autonomous manifold shutdown has been implemented. In the event of thruster failure, either a "hot" fire or "cold fire," the ICM maintains attitude within specification. In the situation of a "hot fire," the fuel usage is great as the system counteracts the perturbation caused by the failed thruster.

Inertial Measurement Units

The ICM has three IMUs–two "hot" units and one "cold" spare. Most Kalman filters used on spacecraft are not classical Kalman filters as described by Kalman but rather they follow the "model-replacement" design described in ref. 2. The problem inherent in the design of ref. 2 is the complete lack of robustness to rate outliers. If the IMU is subject to a single spike of rate shot noise, the effect is to rotate in an unfiltered fashion the attitude estimate of the vehicle. This large erroneous rotation, combined with common "residual gating" (to block measurement outliers) can cause the filter to shutdown the measurement updates and cause the filter to never recover.

To prevent this occurrence, the two IMU's on the ICM perform a series of autonomous cross-checks. The IMU's are rate integrating so they report an angular change every sample period (rather than a rate.) The following checks are then performed:

1. The angular changes of the IMU's are differenced with each other. If the magnitude of this difference is small, then the angular change of both IMU's is forwarded to their respective Kalman filters.
2. If the angular change difference is large, then the angular change of each IMU is "sanity tested," i.e., the angular change is compared against a maximum value based upon the typical rates that the ISS might be subject to. If one of the IMU's fails this test, the failure is noted and the angular change of the "good" IMU is fed to both Kalman Filters.
3. If sanity test passes, then the current angular change of each IMU is compared against its previous value. This sample-to-sample difference is compared against a value representing the typical ISS accelerations.
4. In addition to these tests, a "stuck IMU" test is performed on the IMU output words. Even if the ISS is not rotating, the output words should change due to noise. If they are the same for some extended period, it is presumed that the IMU is malfunctioning.

Given the complexity of these types of cross-checks, it might be of interest to review the application of the model-replacement Kalman Filter to spacecraft since the classical Kalman filter provides immunity to large outlier rate disturbances. Use of nonlinear median filters on the rate would provide a robust method of outlier rejection and should also be further researched.

Star Cameras

Two three-axis autonomous STar Cameras were implemented on ICM to improve attitude determination reliability, to provide improved attitude determination performance (the STC's are not co-aligned), to minimize sun-earth-moon obscuration, and to provide good star visibility.[4]

In the early designs of the attitude determination system, STC selection for the Kalman filter update was autonomous, based upon the calculated positions of the sun, earth, and moon relative to the STC boresight. It was difficult for the camera manufacturer to characterize with certainty the angular size of the "keep-out" cones of these celestial objects. It was therefore decided to rely on the reported performance of the star camera algorithm. If a camera reported a failed star identification or failed attitude fix after three attempts, the attitude determination system would select the alternate STC. If this STC failed after three attempts, the system would "ping-pong" between the trackers until an attitude fix was achieved. There was an Kalman filter update outage during this search and the attitude estimate degraded gracefully according to the statistics of the IMU angular random walk and bias instability.

Four Computer Redundant System

In the early stages of the ISS-ICM project great concern was placed upon computer reliability, particularly regarding potential failures during proximity operations. The original design implemented two computers. The exact implementation of the two computers on ICM was not fully developed.

Two ensure greater reliability and to minimize potential thruster misfirings, a four computer system was implemented. The implementation consisted of two strings of two computers. The 14 bit thruster fire word from each computer in the string is bit-wise 'AND'ed" and then the outputs from each string is bit-wise "OR'ed" to yield a bit-mapped message to the thrusters.

This AND/OR implementation of four computers reduces the probability of an improper thruster firing (hot failure) caused by a hardware failure or a single-event upset (SEU).[5] The probability of a hardware failure or SEU leading to a thruster not firing (cold failure) *is increased* in comparison

[4] Star visibility was a major concern driving the design of camera placement and orientation. As the ISS increases in size during the construction phase, avoiding structural obscuration of the STC's was a difficult problem.

[5] The four computers each has the same software so the probability of a *software-generated* misfire is not reduced by this redundant implementation.

to a two-computer system. This increased probability is because "ORing" the "ANDed" computer strings creates a reliability diagram of two serial systems in parallel. This system has four ways in which a two-computer failure can prevent the system from issuing a thruster fire command. Thus a trade-off was made to reduce the probability of a hardware-failure induced firing at the expense of an increased probability that such a failure woould increase the probability of an RCS system shutdown.

ALGORITHM DEVELOPMENT, ANALYSIS, AND SIMULATION

The entirety of the ICM control system development was carried out by two-to-four engineers on an approximate one year schedule. Development was expedited by utilizing previous experience of the analysts in reaction jet systems and the use of pre-existing software tools developed by those analysts.

ACS Algorithm Development

Preliminary analyses were carried out using MATLAB$^{(R)}$ and the Control System Toolbox to develop control laws on a single axis basis and verify basic rigid body stability. It has been the experience of the authors that MATLAB simulations are useful for rapid proof of concept, but are too slow for production runs due to the interpretive nature of the language. Therefore, the thruster control laws were then coded in Fortran 90 and integrated into the high fidelity simulation module described in the subsequent section.

Simulation

MACSIM32$^{(R)}$ (Multi-body Attitude Control System SIMulation) was developed by Applied Systems Engineering, Inc. and used to characterize ACS performance in all flight modes. The simulation includes models of rigid and flexible body dynamics, sensors and actuators, on-orbit environmental disturbances, ground commands and telemetry, and a unique graphical user interface (GUI) which allows analysts to rapidly study and diagnose anomalous behavior. MACSIM32 is coded in Fortran 90 which is the simulation language of choice in ISS development at NRL and Johnson Space Center (JSC), and thus mandated pre-existing standard environmental and flexibility models are readily integrated. MACSIM32 runs on a desktop PC under Windows 98/NT, and its core modules can be expeditiously configured to study a new spacecraft ACS. This rapid prototyping capability was essential to meeting the schedule requirements of the ICM program. The simulation was adapted to ICM flight dynamics, a controller designed, and performance verification runs made; completed in less than six weeks in time to support the PDR. A functional block diagram of MACSIM32 is shown in Figure 6.

A fundamental design feature of MACSIM32 is the utilization of a pseudo flight software module with interfaces to hardware device models and ground command emulating those of the on-board system. The flight software module and its interface specification could be easily separated and sent to JSC where it was integrated and validated on an entirely independent Fortran simulation. Portability of MACSIM32 on a Pentium Laptop also allowed NRL personnel to travel to JSC with a fully functional simulation when the need arose for side-by-side comparison runs.

Flight Software Algorithm Specification

In order to facilitate flight software development, a 250 page Algorithm Definition Document (ADD) was created to serve as a specification for all real-time modules. Each section of the ADD corresponds to a software module that must be coded, integrated, and tested in the on-board system. The ADD was reviewed internally at NRL and independently by JSC and ultimately was entered into configuration management.

Development of the ADD was facilitated by using the flight software segment of MACSIM32 broken down into individual subroutines forming the basis for the flight module specifications. The actual analytical code was attached to each section as an appendix. Flight software developers were later able to translate the analytical code into flight code on an almost 1 for 1 basis, allowing rapid progression from coding to testing. Testing on the module level was also obviously facilitated by the similarity between the analytical code and the flight code. Static and dynamic test cases could be run on both sets of software and compared, which allowed ready identification of errors.

The importance of the extensive documentation in this large program cannot be overemphasized. In a project the size of the ISS there are a multitude of organizations spread across the country which analyze various aspects of the system design. While the nominal "client" of the NRL ACS group was the NRL software group, many different organizations required the algorithms to further their analyses. With the detailed ADD and MACSIM32 available on the web, it was possible for all the disparate organizations to move forward, more-or-less in lockstep.

CONCLUSION

The proverbial saying that "a camel is a horse designed by a committee" necessarily applies to a project so large as the ISS. While the ICM probably has its share of superfluous camel "humps," it remains nonetheless a success story by every metric. We can learn from the humps and next time try to straighten the camel's back. *Experience* and *design for simplicity* are key to getting the job done right the first time. The project is a testimonial to the achievement possible when dedicated and *experienced* engineers work together to do the job right.

REFERENCES

[1] M. D. Shuster, "Maximum likelihood estimation of spacecraft attitude," *J. of the Astronautical Sciences*, vol. 37, pp. 79–88, Jan. 1989.

[2] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *AIAA J. of Guidance, Control, and Dynamics*, vol. 5, pp. 417–429, Sept. 1982.

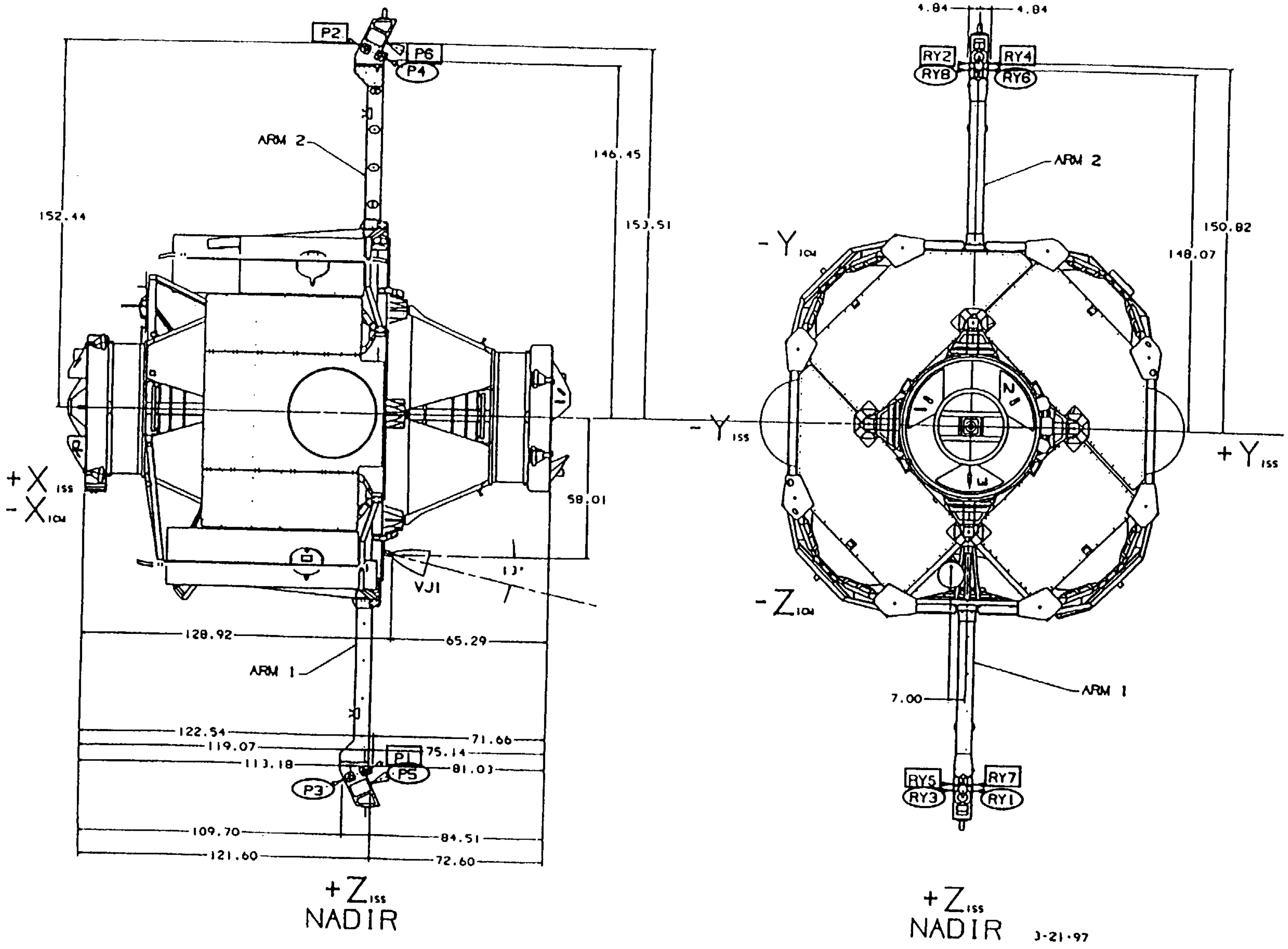


Figure 1: The ICM thruster configuration. Thruster P5, P6 are 110 N (25 lbf) thrusters, the remaining are 22 N (5 lbf) thrusters. Manifolds are geometrically "interlaced." Manifold 1 is represented by the ellipse designations, Manifold 2 is represented by the rectangle designations.

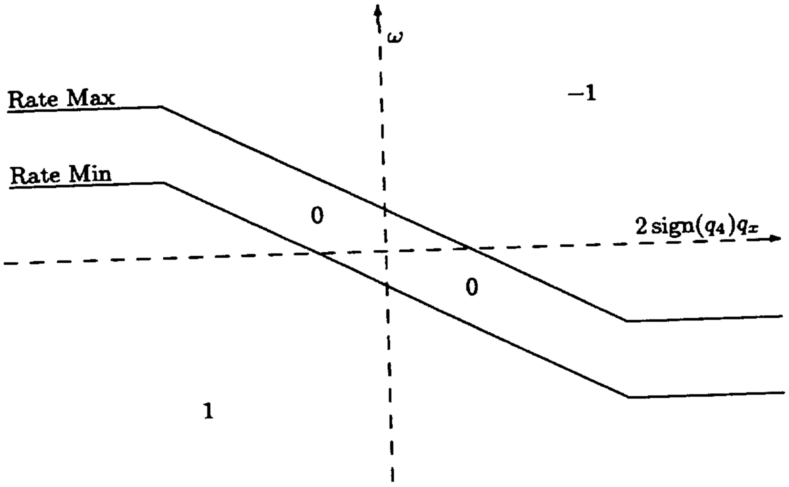


Figure 2: Typical ICM deadband switch curve.

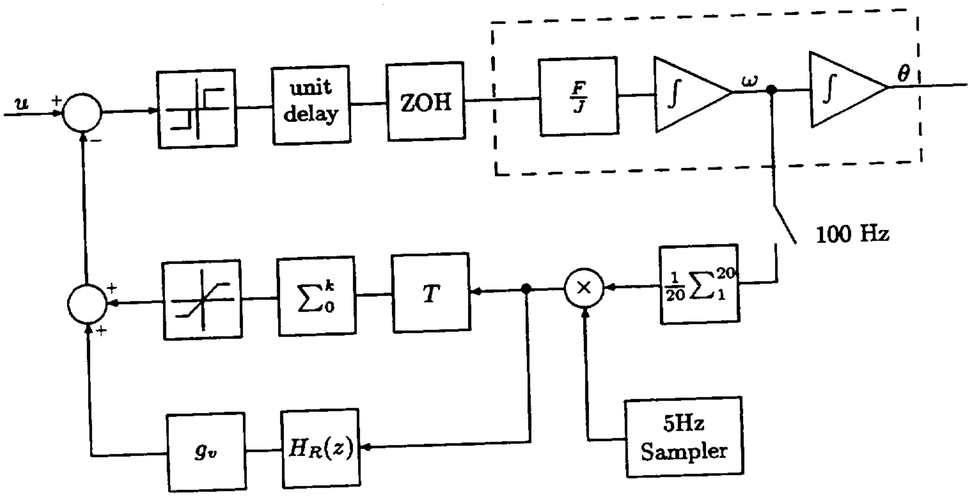


Figure 3: Single-axis block diagram.

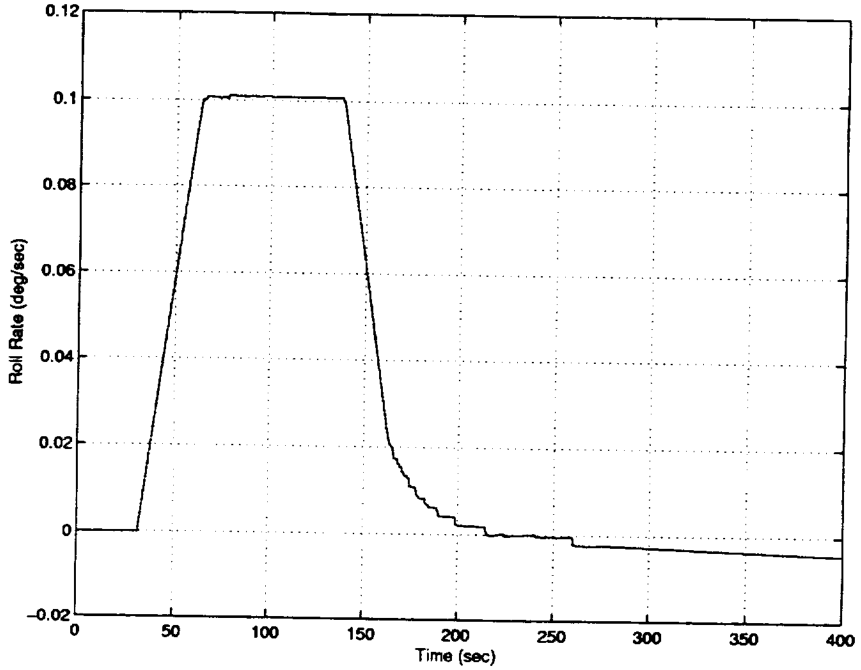


Figure 4: ICM rigid body rate profile for a large angle slew.

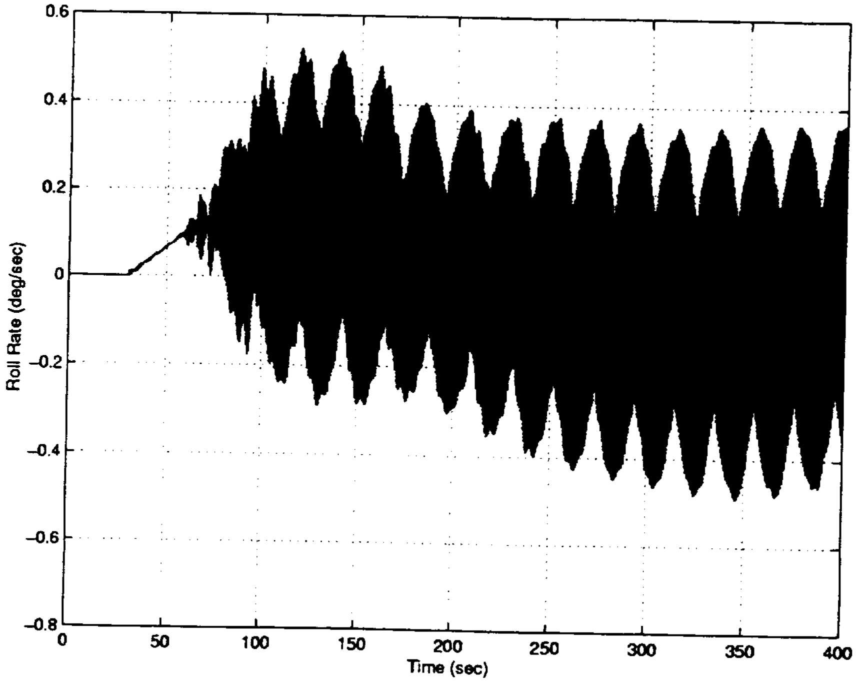


Figure 5: ICM flexible body rate profile for a large angle slew, no rate filter in control loop.

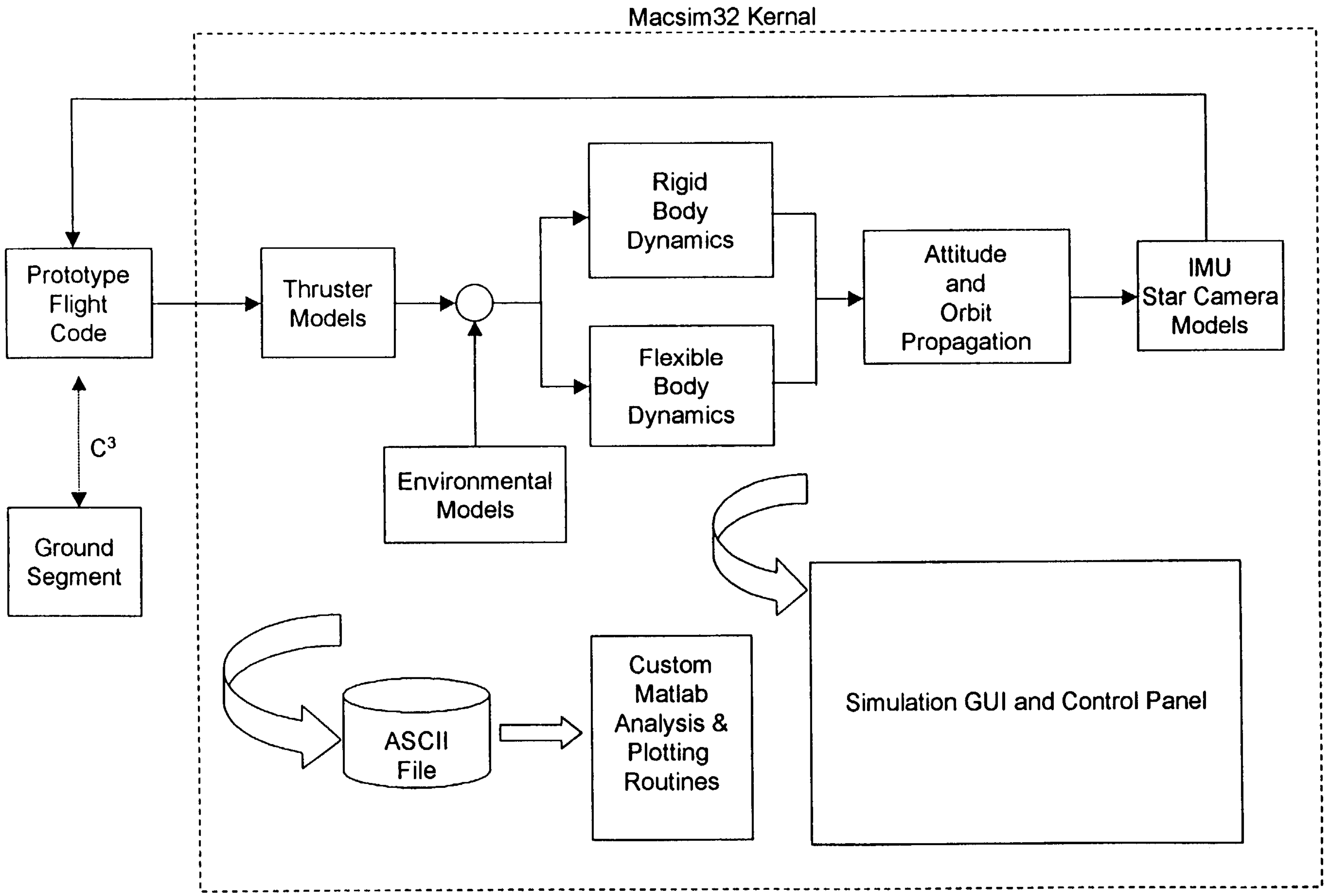


Figure 6: Function block diagram of MACSIM32$^{(R)}$ used to develop and analyze the ICM control system. The simulation evolved from a development tool into program-wide baseline standard. Simulation could run 25× realtime for analysis. Simulation could also be "paced" down to realtime for use during Integration and Test.

BACKUP ATTITUDE CONTROL ALGORITHMS FOR THE MAP SPACECRAFT

James R. O'Donnell, Jr., Ph.D.[1], Stephen F. Andrews[1], Aprille J. Ericsson-Jackson, Ph.D.[1], Thomas W. Flatley, Ph.D.[1], David K. Ward[2], P. Michael Bay[3]

Goddard Space Flight Center
Greenbelt, MD 20771 USA

The Microwave Anisotropy Probe (MAP) is a follow-on to the Differential Microwave Radiometer (DMR) instrument on the Cosmic Background Explorer (COBE) spacecraft. The MAP spacecraft will perform its mission, studying the early origins of the universe, in a Lissajous orbit around the Earth-Sun L_2 Lagrange point. Due to limited mass, power, and financial resources, a traditional reliability concept involving fully redundant components was not feasible. This paper will discuss the redundancy philosophy used on MAP, describe the hardware redundancy selected (and why), and present backup modes and algorithms that were designed in lieu of additional attitude control hardware redundancy to improve the odds of mission success. Three of these modes have been implemented in the spacecraft flight software. The first onboard mode allows the MAP Kalman filter to be used with digital sun sensor (DSS) derived rates, in case of the failure of one of MAP's two two-axis inertial reference units. Similarly, the second onboard mode allows a star tracker only mode, using attitude and derived rate from one or both of MAP's star trackers for onboard attitude determination and control. The last backup mode onboard allows a sun-line angle offset to be commanded that will allow solar radiation pressure to be used for momentum management and orbit stationkeeping. In addition to the backup modes implemented on the spacecraft, two backup algorithms have been developed in the event of less likely contingencies. One of these is an algorithm for implementing an alternative scan pattern to MAP's nominal dual-spin science mode using only one or two reaction wheels and thrusters. Finally, an algorithm has been developed that uses thruster one shots while in science mode for momentum management. This algorithm has been developed in case system momentum builds up faster than anticipated, to allow adequate momentum management while minimizing interruptions to science. In this paper, each mode and algorithm will be discussed, and simulation results presented.

SYSTEM ENGINEERING

The Microwave Anisotropy Probe (MAP) spacecraft, shown in Figure 1, is the second of a series of Medium Explorer, or MIDEX missions (ref. 1). The MIDEX program was designed as an intermediate option between the Explorer Program's full-sized missions and Small Explorer (SMEX) spacecraft. Each class of spacecraft follows a different redundancy philosophy; the full-sized Explorers maximize reliability by implementing fully redundant spacecraft, while the SMEX missions' budget constraints tend to force a single-string philosophy. The MIDEX goal is to use

[1] Flight Dynamics Analysis Branch, Code 572
[2] Systems Engineering Branch, Code 571
[3] Jackson and Tull

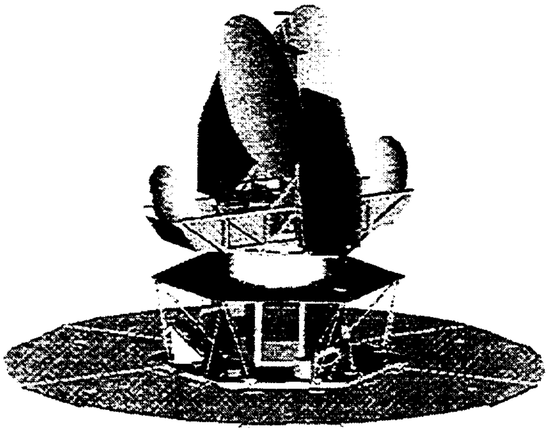


Figure 1: The MAP Spacecraft

redundancy selectively, in order to get the "best bang for the buck", consistent with its mandate of offering the best mission under a fixed cost cap. In many ways, the MIDEX philosophy is the most difficult of the three to follow. There are no set rules for what redundancy should be present, so a decision must be made for each spacecraft element.

MAP's Reliability Philosophy

During the initial system engineering of the MAP spacecraft, and at a few other occasions during the development of the mission (e.g., as a result of points raised at MAP's Confirmation Review), decisions were made as to what redundancy to include, and how to implement it. These decisions were made according to the following reliability selection criteria:

- The overall results of the failure likelihood of hardware components based on historical databases and independent Probability of Success (Ps) calculations were considered.
- Resource impacts of redundant component cost, mass, and power were considered, as well as impacts to the integration and testing schedules.
- Modularity of redundant hardware or software components was evaluated.
- For potential redundant hardware, the availability of algorithmic redundancy (such as the algorithms described later in this paper) was taken into account.
- Conversely, the ability of an additional component to backup multiple components was also taken into account.

MAP Redundancy: Components and Algorithms

As a result of the reliability philosophy detailed above, the following components were selected to have hardware backups. Another transponder was added, based on the failure probability from a historical database and recent experience. Two thrusters were added to the design, due to the criticality of perigee burns and the relatively low cost, mass, and power impacts from the added thrusters. Because the star tracker used on MAP was a complex piece of new technology critical to the success of the mission, a redundant tracker was added. Also, the fact that adding an additional tracker of the same type would be a small modular change to the design, and that both trackers could serve as backup rate sources to the inertial reference units (IRU) made it a good choice for hardware redundancy. A redundant main processor and interface electronics box was added, also because of the complex new technology of the processor used. Finally, an additional set of coarse sun sensors (CSS) was added; there were no concerns about the existing CSSs, but adding a new set was simpler than cross-strapping one set to both interface electronics boxes.

A number of components were not selected for hardware redundancy, even though they appeared high on the historical database of potential failure items. A fourth reaction wheel assembly (RWA) was not added because of mass and power limitations, positive life test experience on the MAP wheel design, and the existence of a possible backup algorithm (which is described later in this paper). A third two-axis IRU, which would provide full redundancy in all three axes, was not added, primarily because of the existence of DSS and star tracker backup algorithms. Redundant power system electronics (PSE) was not added because the existing design was robust enough to survive many failures using existing electronics.

One component was included on the MAP spacecraft even though, strictly speaking, it was not needed at all, in seeming violation of the intelligent redundancy philosophy. The DSS was not needed for any nominal mission mode. It was included on the spacecraft—in fact, a second DSS head was added well after the initial design—because of several factors. The DSS unit is very reliable, and can act as a backup to the two IRU axes which are not already redundant. With nominal mission attitudes, information from the DSS is always available; the second DSS head added allows the unit to be used as a backup rate source during the entire perigee pass (during which MAP must thrust along its own velocity vector). This became important when concerns were raised about poor star tracker performance in the Van Allen radiation belts.

ON-BOARD BACKUP ALGORITHMS FOR MAP

The development of backup attitude control algorithms for the MAP spacecraft, and the decision of which to implement on-board and which to keep in reserve (not implemented but available in the event of in-flight failures), was dictated by the reliability philosophy described in the previous section. In general, those algorithms were placed on-board that were easy to implement and/or provided a backup to a more likely failure item.

Three algorithms of this nature were implemented on MAP. The first allows the on-board Kalman filter to be run with a DSS-derived rate in place of an IRU rate. The second uses the attitude quaternion and a derived rate from MAP's star tracker(s) to be used in place of both IRU rates and propagated attitude. Finally, a means of establishing a bias attitude command with respect to the sun line was implemented; this bias would be used in an algorithm for providing orbit and momentum management using solar radiation pressure.

Kalman Filter w/DSS Rate

Nominally, the MAP on-board Kalman filter uses two two-axis IRUs to provide three axes of rate information, and updates the propagated attitude and the gyro drift bias corrections based on star tracker attitude quaternions and DSS sun vector measurements. In the event of the failure of one of the IRUs, an alternate rate source for either the x or y axis is needed. In this case, it is possible to substitute a DSS-derived rate for the missing IRU axis, and to continue to run the Kalman filter, updating the attitude and gyro bias with star tracker measurements only.

DSS Rate Algorithm

The data from the DSS can be used to compute the body rates about the x and y spacecraft axes. The DSS output is the measured sun vector in the body frame. Given $\hat{s}$, the unit vector to the sun in the body-fixed frame, and $\vec{\omega}$, the body rate, the rate of change of $\hat{s}$ is:

$$\dot{\hat{s}} = -\vec{\omega} \times \hat{s} = \hat{s} \times \vec{\omega}$$

In component form, this can be written as:

$$\begin{aligned} \dot{s}_x &= s_y\omega_z - s_z\omega_y \\ \dot{s}_y &= s_z\omega_x - s_x\omega_z \\ \dot{s}_z &= s_x\omega_y - s_y\omega_x \end{aligned}$$

In the event of a single IRU failure, the body z-axis rate will still be known, because each IRU has one measurement axis in that body axis. With ω_z known, ω_x and ω_y can be found from the DSS measurement by solving the first two component equations for the unknown body rates:

$$\begin{aligned} \omega_x &= (\dot{s}_y + s_x\omega_z)/s_z \\ \omega_y &= (-\dot{s}_x + s_y\omega_z)/s_z \end{aligned}$$

Since the IRU measures the average rate over the previous control cycle, the DSS data must be averaged also, or there will be a half-cycle (0.5 second) offset between the rate that ω_z represents, and the sun vector that s_x and s_y represent. At low rates, this is not a problem, but at the nearly 3°/sec MAP fast spin rate, this half-cycle offset is too large to ignore. Thus, the sun vector used in the above equations must be the average sun vector in the cycle over which the DSS rate (and IRU rate) is calculated:

$$\begin{aligned} s_x &= 0.5(s_x(k) + s_x(k-1)) \\ s_y &= 0.5(s_y(k) + s_y(k-1)) \end{aligned}$$

where k indicates the current cycle, and $k-1$ is the previous cycle. In addition, the sun vector rate must also be the average rate of change of the sun vector over the control cycle:

$$\begin{aligned} \dot{s}_x &= (s_x(k) - s_x(k-1))/\Delta t_{DSS} \\ \dot{s}_y &= (s_y(k) - s_y(k-1))/\Delta t_{DSS} \end{aligned}$$

where Δt_{DSS} is the time between DSS samples.

The end result is the DSS measurement of the x and y axis body rates. Both of these rates are calculated each control cycle and telemetered to the ground, and are available to be used in the Kalman filter. Nominally, the DSS rates are also used in a system rate check that compares the IRU rates, DSS rates, and AST rates to each other.

Kalman Filter Adjustments for DSS Rates

There were two ways that the DSS-derived body rate could be used to replace the IRU rate in the Kalman filter. First, the state equations and noise models could be redesigned to incorporate the different rate source, and that model would be used only when the filter was configured to use the DSS. Second, the DSS rate could be treated as if it came from an IRU, with a parameter change to the existing Kalman filter model. Software impacts and schedule considerations led to the choice of option two. During testing, or in the event of an IRU failure on-orbit, the Kalman filter is configured to use a DSS rate, and the appropriate parameters are adjusted to account for the different statistical qualities of the DSS rate compared to the IRU rate. Although the DSS rate is correlated in time through the calculation of the average sun vector, each DSS-derived rate was treated as if it were uncorrelated. In addition, since the DSS output is a position measurement, there is no drift associated with it, as with an IRU measurement. For these reasons, the standard deviation of the IRU rate random walk (σ_u) was reduced by half to model the decreased drift characteristics of the DSS rate measurement. In addition, since a noisy DSS measurement was being differentiated to generate the DSS rate, this rate would have higher white noise than the IRU rate. Thus, the rate white noise standard deviation (σ_v) was increased by an order of magnitude over the nominal IRU model. Finally, the DSS was disabled as a filter update sensor.

Simulation Results

Figure 2 and Figure 3 show, respectively, MAP's simulated IRU-measured x- and y-axis body rates and the Observing Mode performance (precession and spin rates and sun line angle) when in its nominal configuration. Figure 4 shows what the corresponding DSS-derived x- and y-axis body rates look like. As expected, they are much noisier. (Note that the MAP flight software was designed with the capability to filter both the DSS- and AST-derived rates, though no filter has currently been put in place for either.) Figure 5 shows the simulated Observing Mode performance when the DSS x-axis body rate is used in the on-board Kalman filter.

Unsurprisingly, the Observing Mode performance in the backup DSS rate mode is noticeably degraded from the nominal case. However, both the sun line angle and spin rate are well within their respective specifications of 22.5 ± 0.25° and 2.784°/sec ± 5%. The precession rate requirement of 0.1°/sec ± 5% is violated in this backup mode, though degraded performance would be acceptable in this case. It is possible that a DSS rate filter would improve these results.

AST-Only for Attitude and Rates

Because the star trackers selected to be used on the MAP spacecraft are quaternion-output trackers, it is a fairly straightforward process to derived rates in each body axis from successive attitude quaternions. If the attitude quaternion at a given time k is denoted q_k, then the Δ quaternion from one cycle to the next can be calculated as $\Delta q = q_{k-1}^{-1} \otimes q_k$, where $\otimes$ denotes quaternion multiplication. Since Δq can be expressed as the Euler axis and angle parameters $[e_1 \sin(\Delta\phi/2), e_2 \sin(\Delta\phi/2), e_3 \sin(\Delta\phi/2), \cos(\Delta\phi/2)]$, the derived rates can be found using the relationship $\omega = [e_1\, \Delta\phi/\Delta t, e_2\, \Delta\phi/\Delta t, e_3\, \Delta\phi/\Delta t]$.

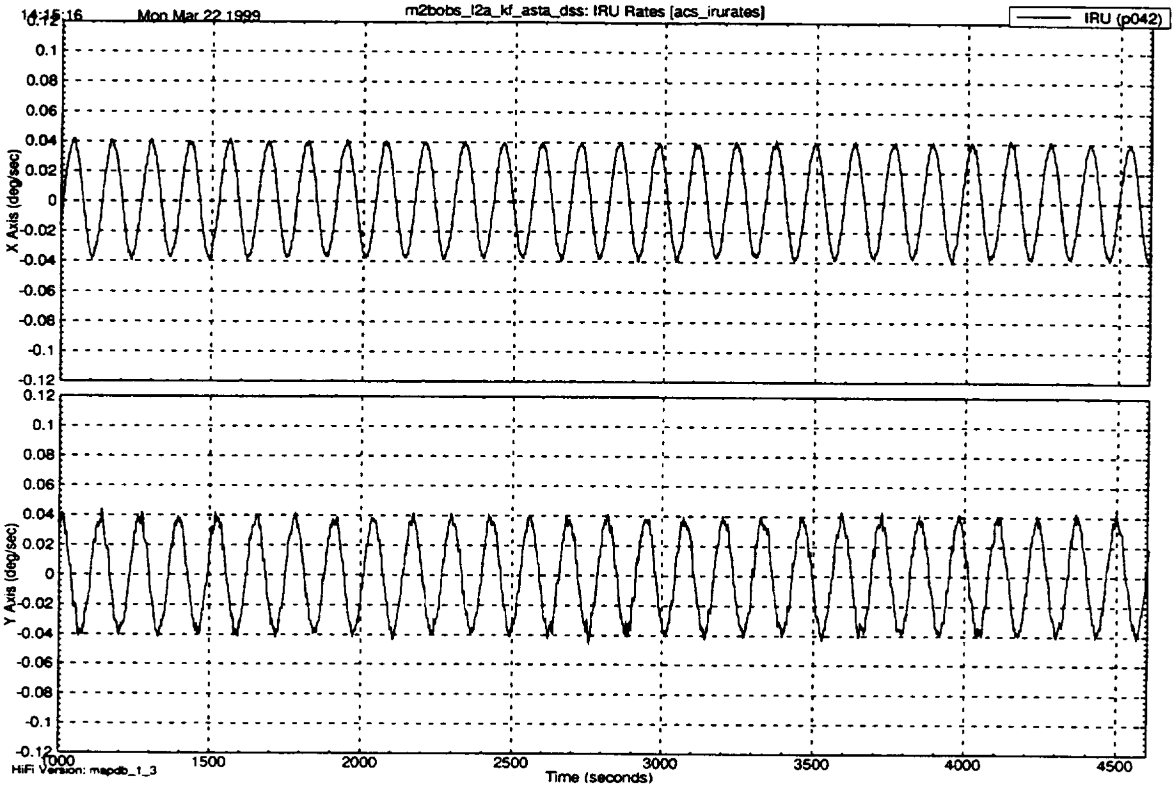


Figure 2: Nominal IRU Body Rates

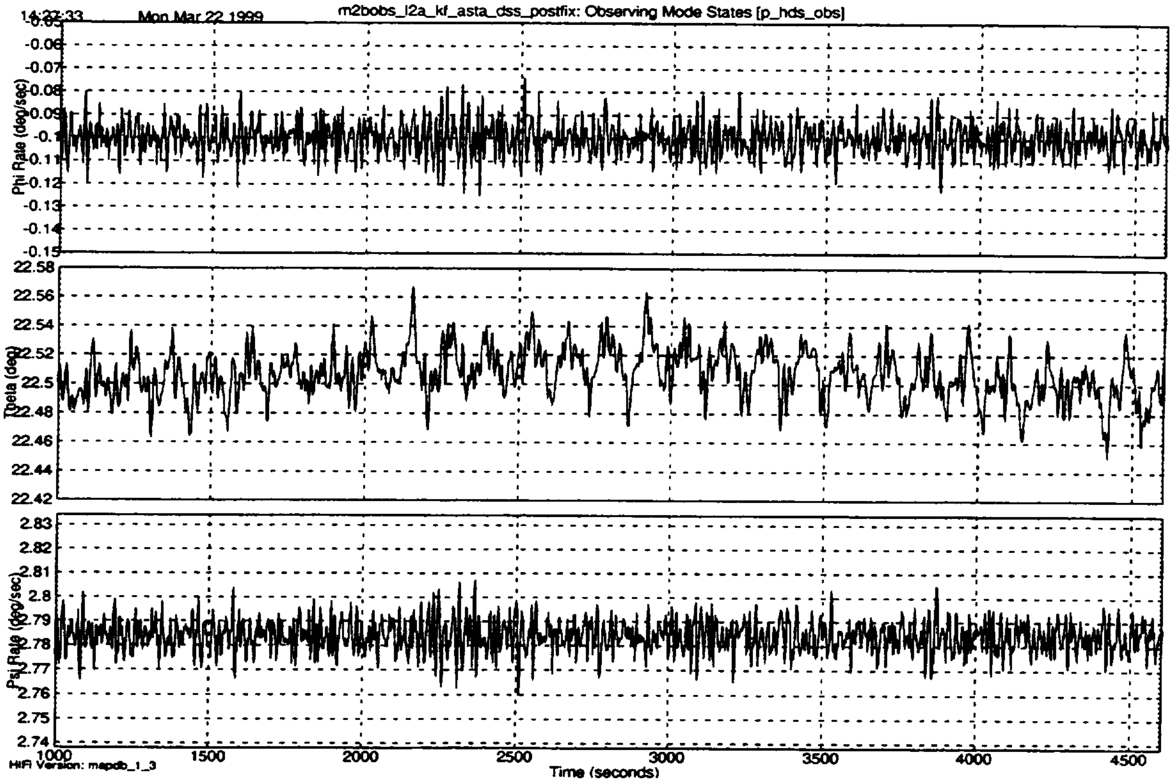


Figure 3: Nominal Observing Mode Performance

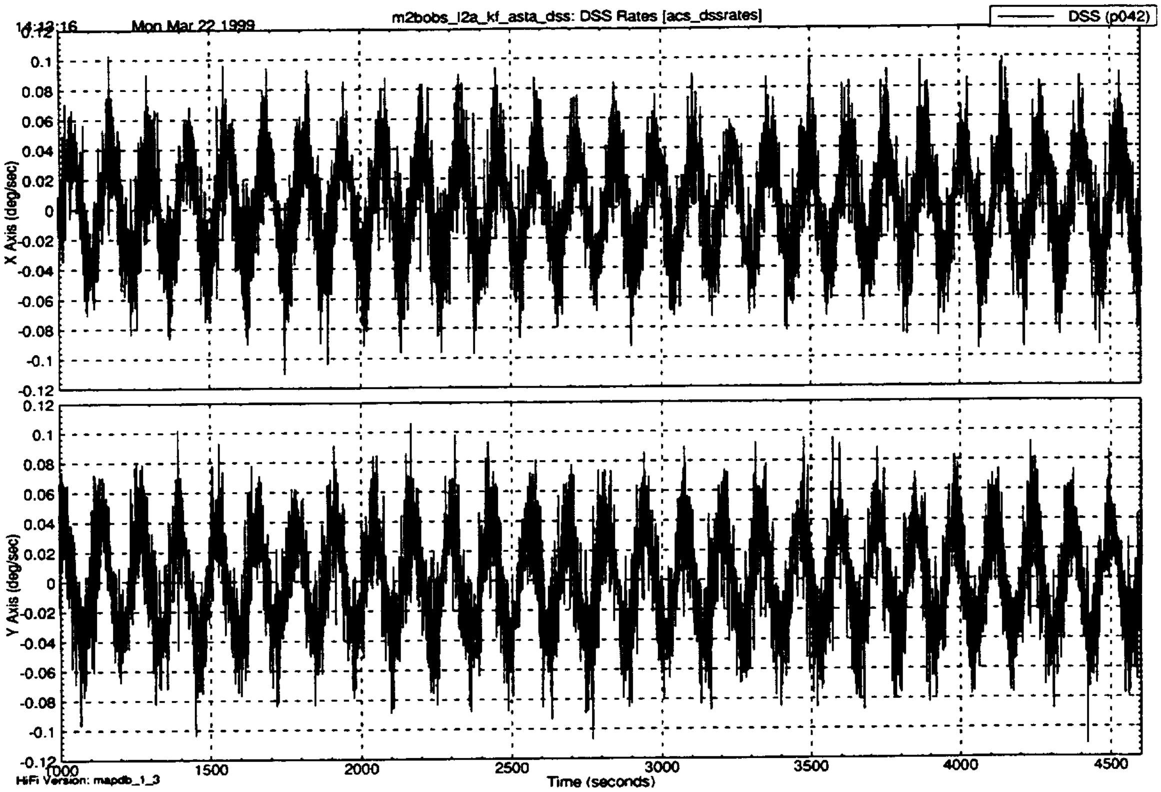


Figure 4: DSS-Derived Body Rates

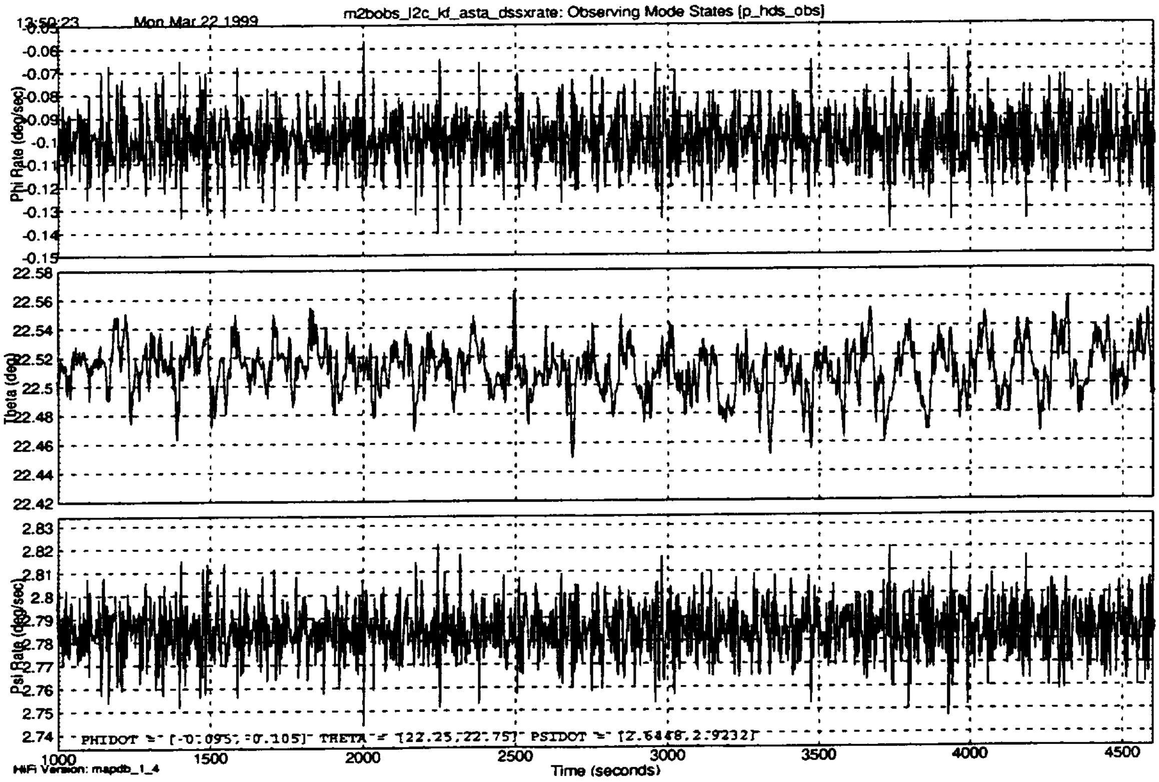


Figure 5: Observing Mode Performance Using DSS-Derived Rate in Kalman Filter

Figure 6 shows star tracker derived x- and y-axis body rates (the z-axis rate shows similar characteristics; only x and y are shown here to make it easier to compare with the IRU and DSS rates shown in Figure 2 and Figure 4). Figure 7 shows the Observing Mode performance using star tracker attitude and derived rates. As with DSS rate backup algorithm, the performance of the system in Observing Mode using the star tracker backup algorithm satisfies the sun line angle and spin rate requirement, but not the precession rate requirement. It is also interesting to note that, while the star tracker derived rates appear to be a bit better than those from the DSS, the Observing Mode performance is slightly better with the DSS rate backup. This is because of the action of the Kalman filter, which can still be used in the DSS case.

Sun Line Bias Algorithm

Solar Radiation Pressure (SRP) will be the main perturbation to the MAP spacecraft orbits when it is in the vicinity of the Earth-Sun L_2 Lagrange point. The two main effects of the SRP on MAP will be to shift the center of its orbit about the L_2 point, and also to cause a buildup of spacecraft system momentum. It is nominally planned to conduct orbit and momentum maintenance operations with the spacecraft four times a year.

Tene, et. al., have proposed a means by which, using small variations in the orientation of MAP's sun shield as it spins and cones about the sun line in its Observing Mode, it might be possible to use the SRP to reduce or even eliminate the need to conduct thruster operations for the purpose of orbit maintenance (ref. 2). Because of the sensitivity of the MAP orbit about L_2, small accelerations are capable of either causing or preventing the spacecraft to escape from the vicinity of L_2.

The average acceleration imparted on MAP due to the SRP was estimated to be 0.2 μm/sec^2. Since the spin axis of the MAP spacecraft precesses about the sun line every hour, the average SRP acceleration is directed away from the sun. If the precession axis points exactly at the sun and the precession motion is symmetric, there would be no acceleration perpendicular to this axis. Any bias in the precession axis would cause a perpendicular acceleration to the sun line; it is this acceleration that Tene proposes could be used for orbit maintenance. During Observing Mode, MAP's spin axis must be precessed about the sun line at an angle of 22.5 ± 0.25°. Given the nominal expected performance of the Observing Mode controller, an offset as high as 0.1° could be applied and still allow science requirements to be met. Tene showed that with such an offset, an average acceleration can be applied perpendicular to the sun line on the order of 0.5 nm/sec^2. With the correct bias, this acceleration can help MAP maintain a desired orbit.

Tene originally proposed an onboard controller that would take attitude and orbit information to autonomously determine the sun line bias to be applied. Subsequently, it was determined that the necessary bias changes slowly enough (on the order of a week or more) that it could be very simply implemented as a constant sun line bias, expressed as a quaternion rotation about MAP's rotating sun reference (RSR) frame, that could be commanded by the ground. This bias would rotate the entire RSR frame; the Observing Mode Euler angles and rates would remain unchanged. However, the bias would show up in the sun angle. Figure 8 shows the effects of a 0.1° sun line bias commanded at 500 sec.

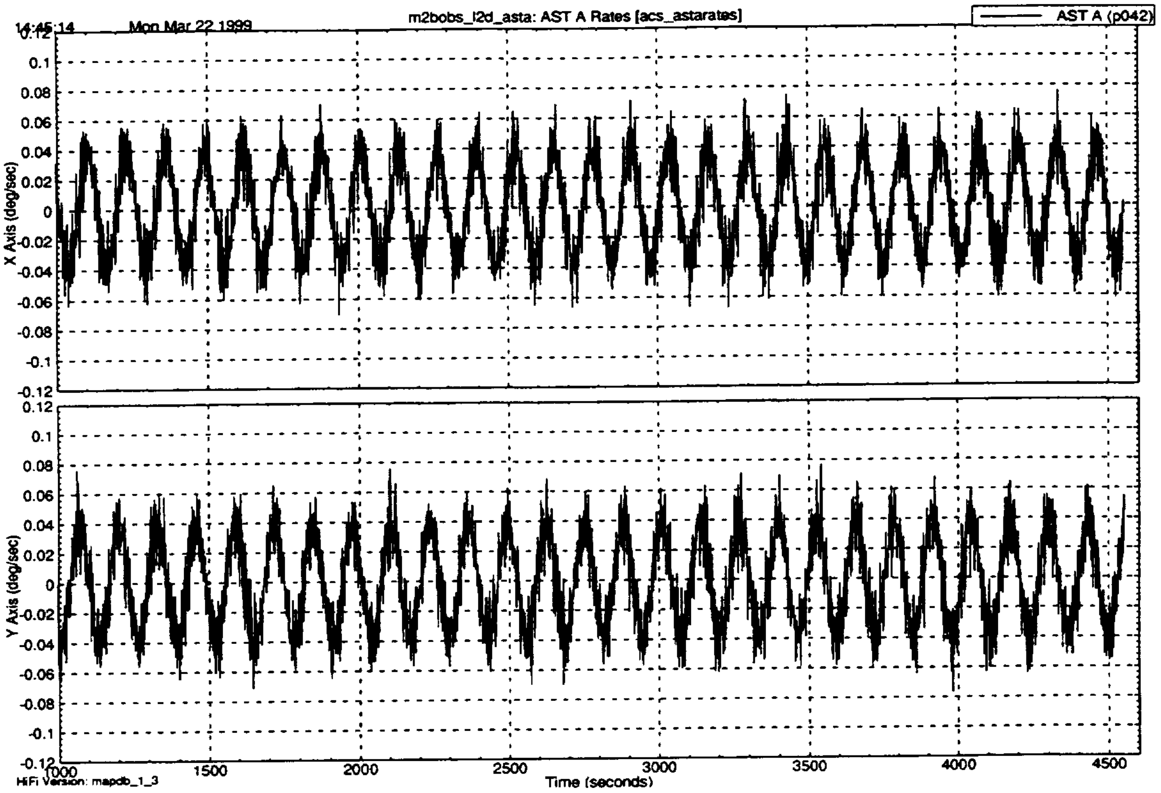


Figure 6: Star Tracker Derived Body Rates

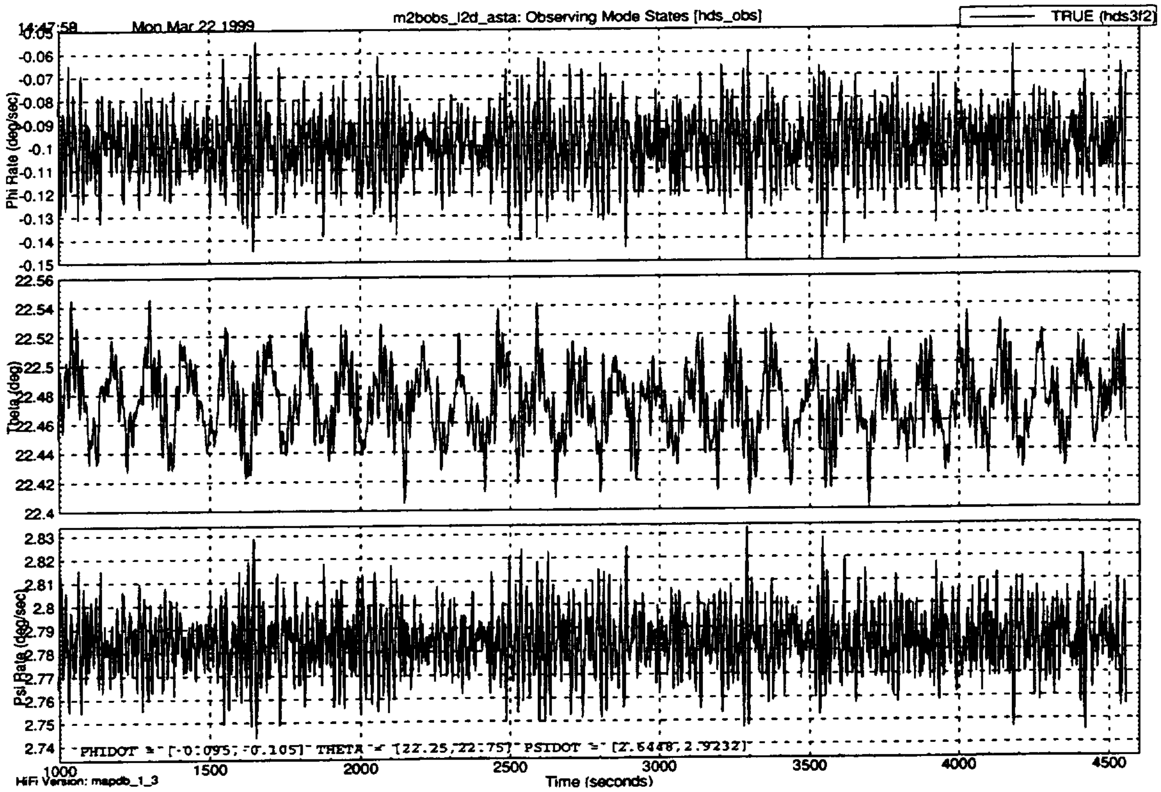


Figure 7: Observing Mode Performance Using Star Tracker Attitude and Derived Rate

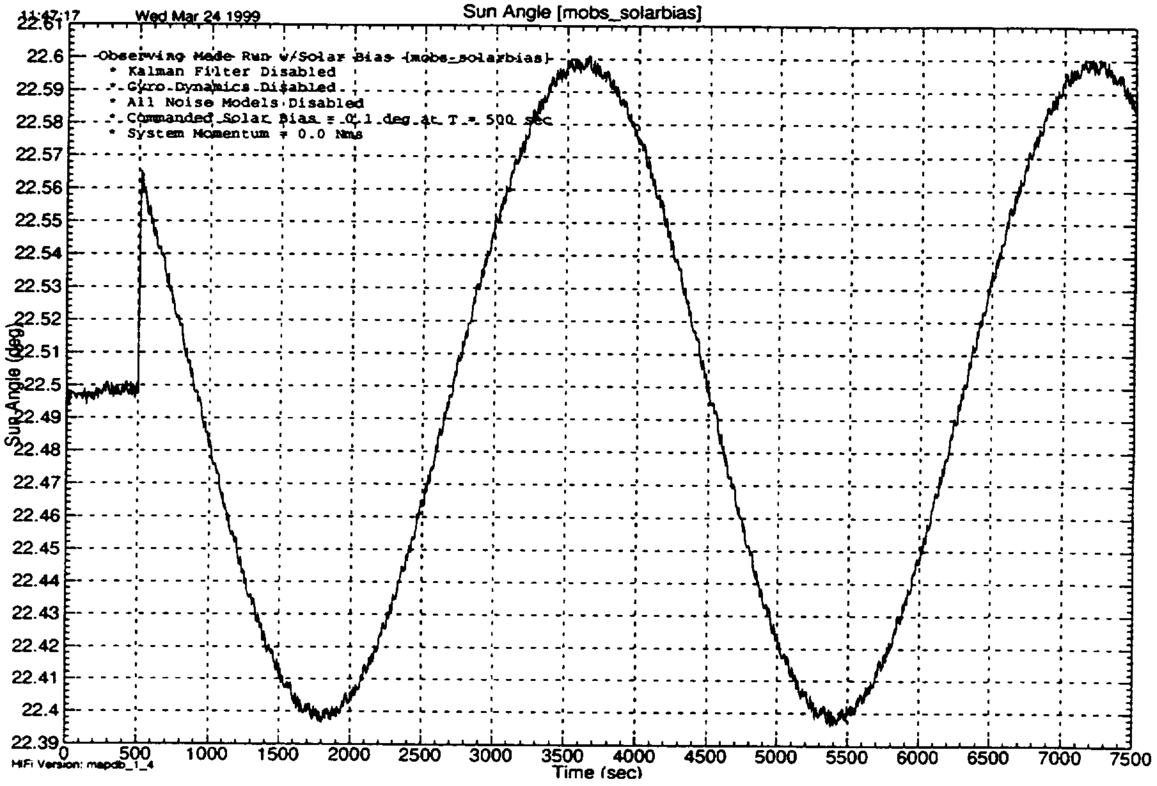


Figure 8: Sun Line Bias Angle

OTHER MAP BACKUP ALGORITHMS

In addition to the onboard backup algorithms implemented in the MAP flight software, a number of algorithms have been developed to cover other eventualities. These algorithms are not slated to be implemented on the spacecraft, either because of their complexity and impact that would have on the rest of the spacecraft flight software, or because the failure mode for which they were designed was considered relatively unlikely. However, the availability of these algorithms means that in the event that they are needed, a flight software patch to implement them can be written, tested, and uploaded without unnecessary delays.

Observing Mode Thruster Unloading

As mentioned in the previous section, MAP is baselined to perform orbit and system momentum management operations four times a year. Using Tene's sun line bias, it might even be possible to reduce this number. However, due to a fairly large uncertainty about the rate of system momentum buildup, caused by a potential pinwheel torque on MAP (ref. 3), it may be necessary to use thrusters to dump momentum more often. In order to minimize the number of disruptions to science operations, an Observing Mode Thruster Unloading backup algorithm has been developed that: 1) unloads momentum to ≤ 0.3 Nms while in Observing Mode, 2) does not violate the 25° sun line constraint (violations of the 22.5 ± 0.25° Observing Mode sun line angle constraint were permissible), and 3) can be completely executed during one ground pass of approximately 37.5 to 45 minutes.

Figure 9 shows a sketch of a "three shot" unloading process, which uses three thruster firings to unload system momentum. The steps in this process are as follows:

A. After the algorithm is enabled, wait until the transverse momentum is all in the +x axis. Fire thruster 2 to remove as much of this momentum as possible. This takes a maximum of one spin cycle (< 130 sec). [Coordinate system: X1,Z1]

B. After thruster firing A, wait until the sun is in the (–x,z) quadrant of the x-z plane. Fire thruster 1 or 2 (depending on the sign of the z-axis momentum) to add x-axis momentum equal to the amount of momentum in the z axis. This takes a maximum of one spin cycle (< 130 sec). [Coordinate system: X1,Z1] Note that this results in an intermediate system momentum state as much as $\sqrt{2}$ higher than the initial value. For a 1.5 Nms initial system momentum, this intermediate state could be as high as ~2.12 Nms. In simulation, this system momentum value does not pose an attitude control problem.

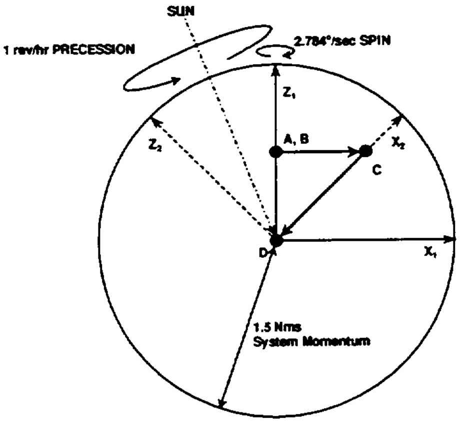


Figure 9: Unloading Schematic

C. After thruster firing B, wait approximately half of a precession cycle, and then wait until all of the system momentum is in the +x axis. Fire thruster 2 to remove as much of this momentum as possible. This takes a maximum of one half precession cycle plus one half spin cycle (< 1865 sec). [Coordinate system: X2,Z2]

D. After thruster firing C, a maximum of 35.5 minutes after the Observing Mode Thruster Unloading algorithm is enabled, system momentum is reduced close to zero. In simulation, it was found that the system momentum was reduced to less than 0.3 Nms for initial system momentum of 1.5 Nms.

Note that the algorithm as described uses thrusters 1 and 2. It could easily be adjusted to use only one of these two thrusters, or to use one of the other thruster pairs.

In simulation, this algorithm fulfills all of the previously stated requirements. Figure 10 shows the three thruster firings used in the example, one from thruster 1 and two from thruster 2. Figure 11 shows the resulting system momentum state of the spacecraft. Note that, after the first firing the system momentum is lower. The second firing, however, is used to align the system momentum vector so that, half a precession cycle later when the spacecraft has changed orientation by 45°, it will appear almost completely in the x-y plane. In this case, this caused a temporary increase in the system momentum of the spacecraft. The third firing reduces the system momentum to below 0.3 Nms. Figure 12 shows the Observing Mode performance during the operation of this algorithm. Other than the brief amount of time following each thruster firing, performance remains within requirements for the entire operation.

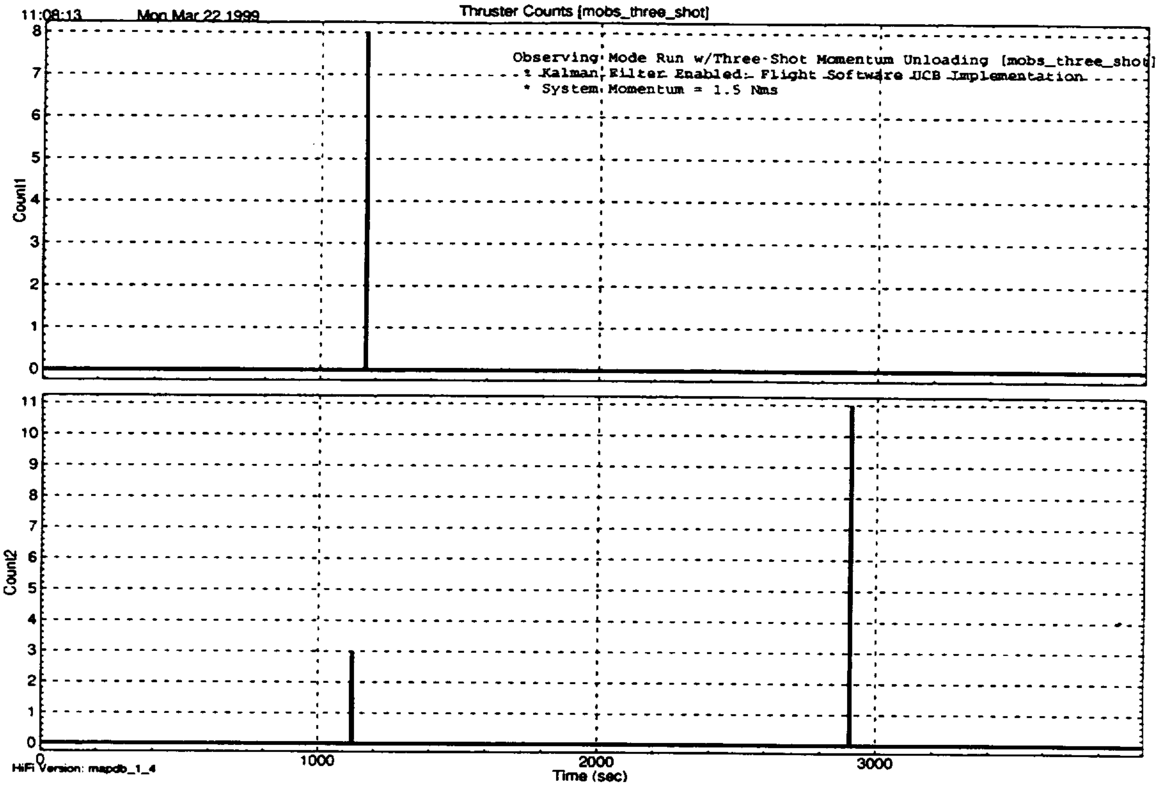


Figure 10: Observing Mode Thruster Unloading

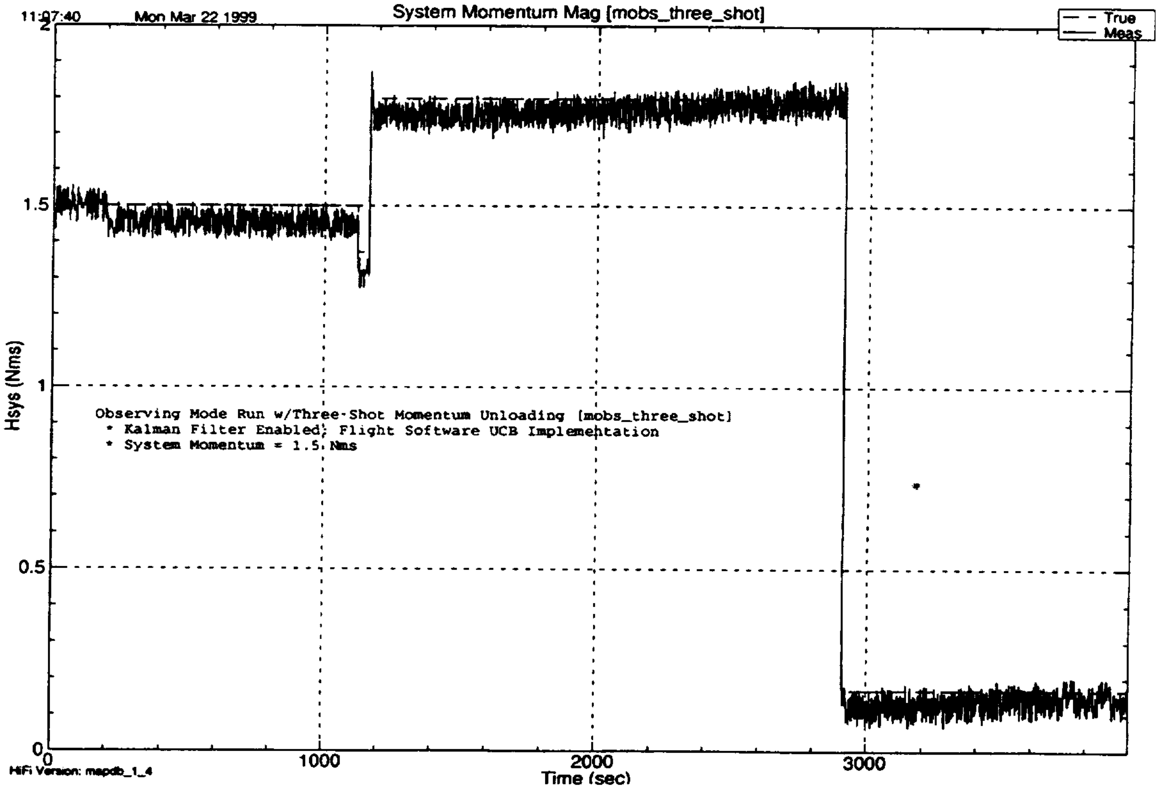


Figure 11: Observing Mode Thruster Unloading System Momentum

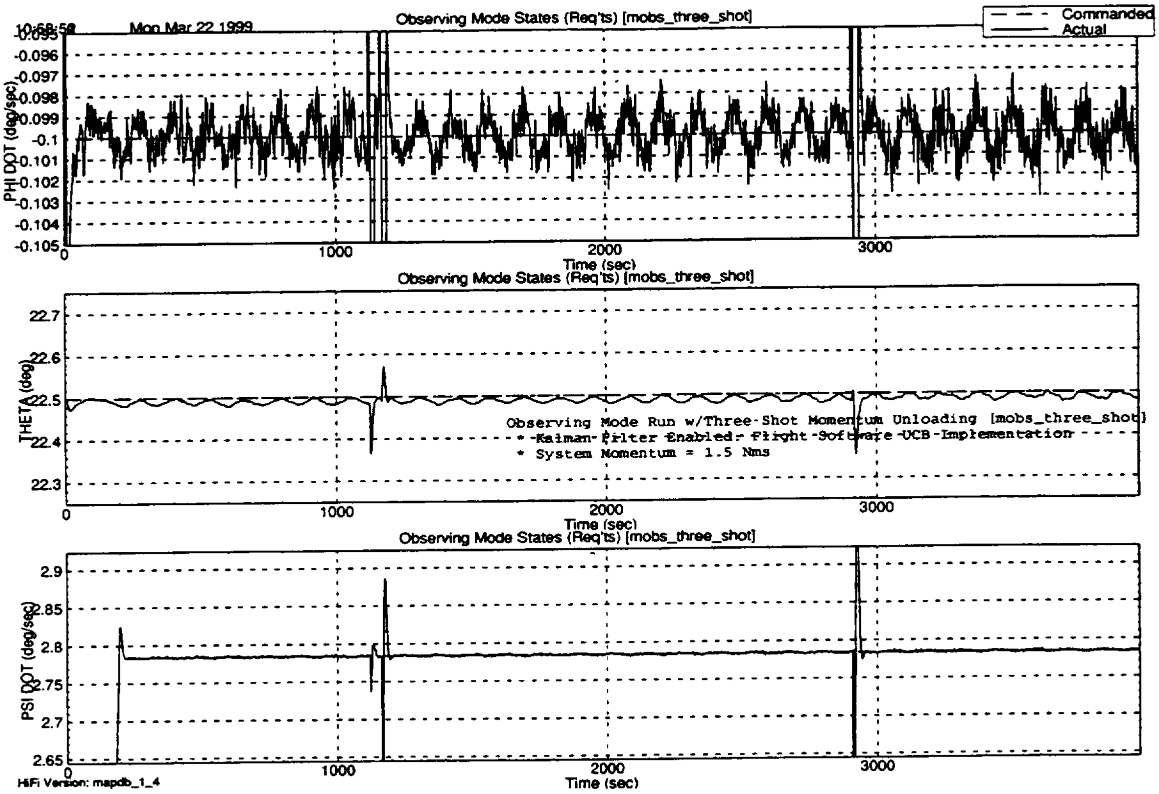


Figure 12: Observing Mode Performance During Thruster Unloading

Observing Mode Using One or Two Reaction Wheels

If one or two of the MAP reaction wheels were not available for control, the spacecraft could be spun up around the z-axis using thrusters. Then, using a single thruster in a pulsed mode (one pulse per spin period) the angular momentum vector could be moved into alignment with the sun vector with a simple spin-axis precession control law. During this maneuver, the spacecraft spin axis would continually nutate around the moving momentum vector. Each pulse would change the nutation (or cone) angle slightly, either increasing or decreasing it. The cone angle buildup would be well-bounded. In the vicinity of the target (the sun), a natural nutational instability would occur and would be taken advantage of. Once the sun is inside the nutation cone, each firing of the jet would increase the cone angle and disturb the angular momentum vector slightly in a pseudo-random direction. The basic motion would approximate a coning motion of the spin axis around the sun line with a continually-increasing nutation angle.

The nutation buildup would cease when the precession law is disabled. The spacecraft spin axis would then be coning around the sun line at some final cone angle. If this angle is 22.5°, the scan pattern produced for the sensors will cover the same region of space in the same amount of time as the normal mode controller, but the detailed motion will be significantly different (see Figure 13). The nominal controller provides a slow precession rate and fast spin rate. With the nutating single-spinner the precession rate is actually faster than the spin rate.

Since the spacecraft spin-axis is its maximum moment of inertia axis, the nutation angle will tend to decrease as a result of energy dissipation in the system. One reaction wheel could be used to maintain the nutation angle at a desired magnitude with a simple control law. Since the sun will tend to drift away from the angular momentum vector, periodic jet firings would also be required to keep this deviation within acceptable limits.

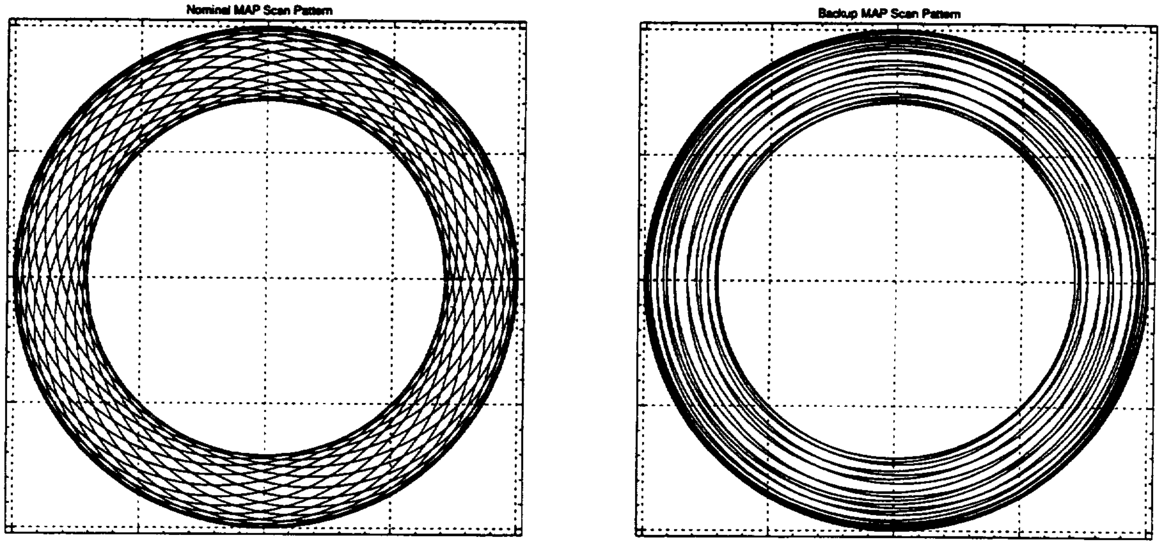


Figure 13: Nominal and Two-Wheel Backup Mode Scan Patterns

Figure 14 shows the sun angle in the body using this control algorithm. At 100 seconds, the spacecraft is spun up with a z-axis thruster firing. From 1000 to 3600 seconds, a single reaction wheel is used to control the nutation angle, bringing it to its steady state value by about 3200 seconds. Beginning at 3600 seconds, single-pulse thruster firings are used to align the spacecraft system momentum vector with the sun vector in the body frame. As described above, once the sun is within the nutation cone (after 4000 seconds in this example), further thruster firings disturb the system momentum vector in a pseudo-random direction and cause the nutation angle to increase. At 6000 seconds, the thruster firings are discontinued and the single-wheel nutation controller is enabled, used to minimize the nutation angle and keep the spacecraft precession cone angle within some tolerance of the desired value of 22.5°. (The residual nutation angle after the system has reached steady state is a result of the system momentum vector not being perfectly aligned with the sun vector.) After 8000 seconds, the spacecraft settles into the dual spin motion that will give the scan pattern shown in Figure 13.

CONCLUSION

In this paper, we have discussed how a philosophy of selective redundancy was applied to the MAP spacecraft. The algorithms that have been designed for MAP, both those currently implemented in the spacecraft flight software and those held in reserve, allowed the project to modify the traditional full redundancy philosophy without a major degradation on the mission's probability of success. These algorithms allowed the project to focus its programmatic resources on other components that do not lend themselves to algorithmic workarounds, improving the reliability of the mission while maintaining the cost capped budget.

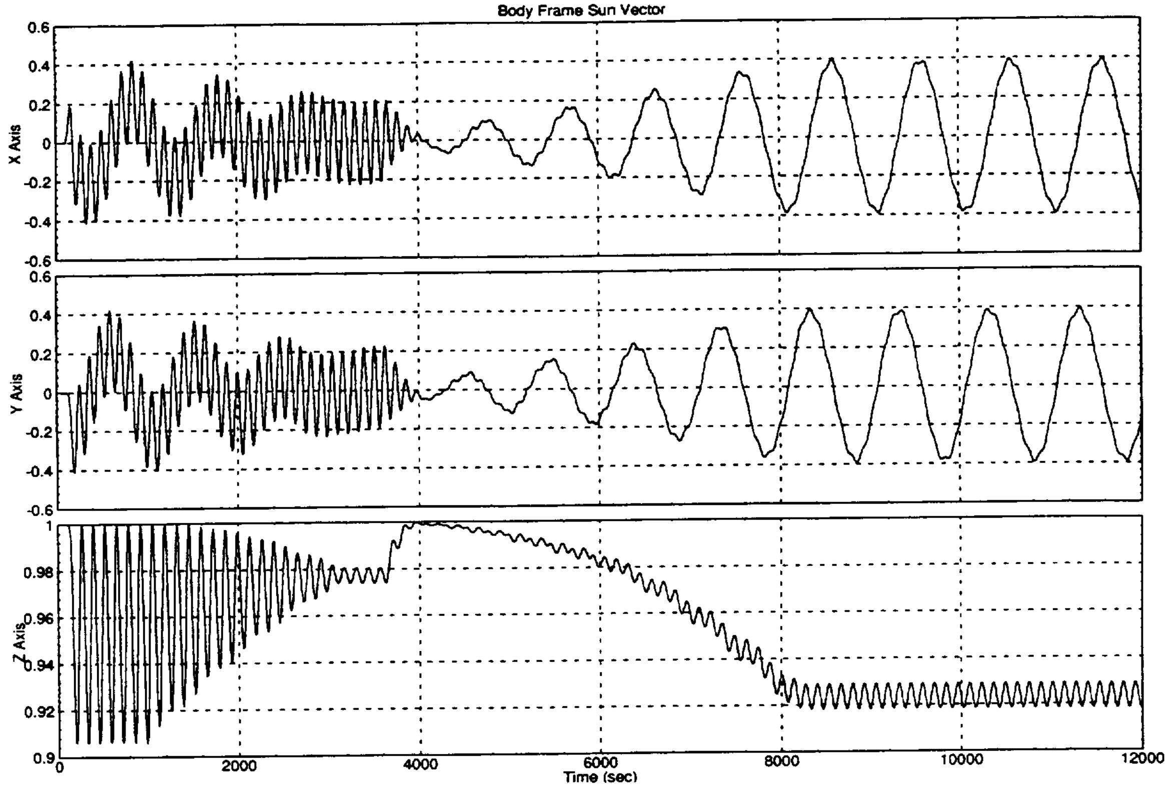


Figure 14: Two-Wheel Backup Mode Sun Vector in the Body Frame

DEDICATION

This paper is dedicated to our mentor and friend, Tom Flatley, who passed away late last year. Thanks to Tom's intelligence, creativity, and perseverance, several missions have enjoyed extended lifetimes despite on-orbit failures, by using algorithms similar to the ones described in this paper. Members of the space community will miss Tom's contributions to our profession through his simple, elegant designs, conceived within a mastery of flight dynamics. Members of the NASA Goddard community will also miss his quiet, unassuming dignity and warm, witty friendship.

REFERENCES

[1] Andrews, S., C. E. Campbell, A. J. Ericsson-Jackson, F. L. Markley, and J. R. O'Donnell, Jr., "MAP Attitude Control System Design and Analysis," *Flight Mechanics Symposium*, Goddard Space Flight Center, Greenbelt, MD, May, 1997.

[2] Ericsson-Jackson, A. J., S. F. Andrews, J. R. O'Donnell, Jr., and F. L. Markley, "MAP Stability, Design, and Analysis," *13th International Symposium on Space Flight Dynamics*, Goddard Space Flight Center, Greenbelt, MD, May, 1998.

[3] Tene, N., Karen Richon, David Folta, and Kimberly Tene, "Using Solar Radiation Pressure to Control L_2 Orbits," *13th International Symposium on Space Flight Dynamics*, Goddard Space Flight Center, Greenbelt, MD, May, 1998.

AN OBJECT ORIENTED SIMULATION ARCHITECTURE FOR RAPID SPACECRAFT PROTOTYPING

Robert R. Strunce, Jr.
Star Technologies Corporation

Fran A. Maher
Independent

ABSTRACT

This paper describes the Object Oriented Simulation Architecture for Rapid Spacecraft Prototyping. The design of this Spacecraft Dynamic Simulator Architecture incorporates a uniquely designed C++ Virtual DataBase Class with linked and embedded Classes for the Environment, Spacecraft (including dynamics), Sensors, Actuators, Time, Automated Procedures, Visualization Windows and Communication Sockets for distributed processing. This architecture provides the foundation for prototyping spacecraft without the need for re-compiling for each variation of the spacecraft design mission. This architecture lends itself to both real-time and non real-time simulations where it may be used in the (1) Attitude Determination and Control System (ADACS) design process; (2) Flight Software Unit Testing; (3) Flight Software/ Hardware-In-The-Loop(HIL) real-time Integration & Test (I&T); (4) Ground Control System for mission planning and checkout; (5) Post Flight anomaly investigation. The Heritage for the simulation models is the GSFC Hybrid Dynamic Simulator(HDS) which was converted to C++ Object Oriented Code. The significant features are: (1) the Virtual DataBase(VDB); (2) Autonomous State Integration; (3) Transformation Engine; (4) Scripting Language.

INTRODUCTION

The Object Oriented Design (OOD) approach to Spacecraft Dynamic Simulation Architecture employs Windows NT Operating System which pulls together several standard models of key spacecraft subsystem, and combines them, in a Class Structure, with existing orbital and space environmental models to produce a very complete, yet re-configurable spacecraft prototyping tool. The C++ OOD code addresses development of OOD Classes for inertial reference systems for stabilization and navigation, coarse sun sensors, fine sun sensors for orientation and navigation, three axis magnetometers, earth sensors for orientation, star tracker, and actuator models for stabilization and maneuvering (reaction wheels, magnetic torque coils/bars, and thrusters). OOD Classes are developed for virtually all of the relevant space environment and orbital parameters of interest including ephemeris models of the sun and moon, solar models , earth occultation models, near earth magnetic fields, near space atmospheric density and drag, internal and external system disturbances, gravity gradients, sensor/actuator disturbances, with rotating antennas and solar arrays. The OOD architecture has provision for incorporating new OOD Classes such as an Electrical Power Subsystem to easily expand the simulation capabilities.

The advantages of this architecture are that the development of a complete spacecraft may be accomplished in just a few lines of code where all the data is encapsulated within each Object, all variables are attached to a Virtual Data Base Object, and the dynamic models are associated with the StateVector integration function as:

```
CSS[0] = new CoarseSunSensorAssembly(3);       3 coarse sun sensors in a block called "CSS[0]"
DSS[0] = new DSSAssembly(1);                   1 digital sun sensor in a block called "DSS[0]"
TAM[0] = new MagnetometerAssembly(3);          3 magnetometers in a block called "TAM[0]"
IRU[0] = new GyroAssembly(6);                  6 gyroscopes in a block called "IRU[0]"
ESA[0] = new EarthSensorAssembly(2);           2 earth sensors in a block called "ESA[0]"
MTB[0] = new MTBAssembly(&bFieldIF,3);         3 magnetic torque bars in a block called "MTB[0]"
RCS[0] = new ThrusterAssembly(4);              4 RCS thrusters in a block called "RCS[0]"
RCS[1] = new ThrusterAssembly(4);              4 RCS thrusters in another block called "RCS[1]"
RWA[0] = new ReactionWheelAssembly(4);         4 reaction wheels in a block called "RWA[0]"
```

These Objects are then attached to a Spacecraft Structure Class. To create a constellation of 60 spacecraft, one only has to instantiate the Spacecraft Class 60 times or once for each spacecraft. The result would be 60 independent Spacecraft Objects each with all its own appropriate encapsulated data and dynamics.

STATE VECTOR / VIRTUAL DATA BASE

As shown in Figure 1, the StateVector is formed which will be integrated over a desired integration time step consistent with the dynamic models in the simulation sensors, actuators or subsystems. The StateVector Class performs the integration function over one time step at the statement containing the code "StateVector++". Transparent to the User, the StateVector Class searches for all instantiated dynamic models, See Figure 2, and through each models derivative function obtains the model derivatives(accelerations, velocities, positions) which are added to the overall state. After the integration function has been performed, the current state of each derived parameter is placed into the appropriate model. In addition, a processing function is called which determines the frequency of the processing thereby providing a muli-rate capability for any instantiated model. The StateVector Class autonomously changes size depending on which dynamics models are turned on or off. The current architecture includes the appropriate orbital parameters of interest As the physical models such as sensors or actuators are instantiated, the encapsulated model parameters are transparently added to the VDB structure, Figure 1, where they can be viewed by clicking on the system level, subsystem level or any level to open the structure as shown in Figure 1 all the way down to the lowest level. This is similar to opening files , folders, and subfolders until you fine the file you are looking for. For example one can see the mass of the spacecraft by open levels SC->MassProperties->Mass.

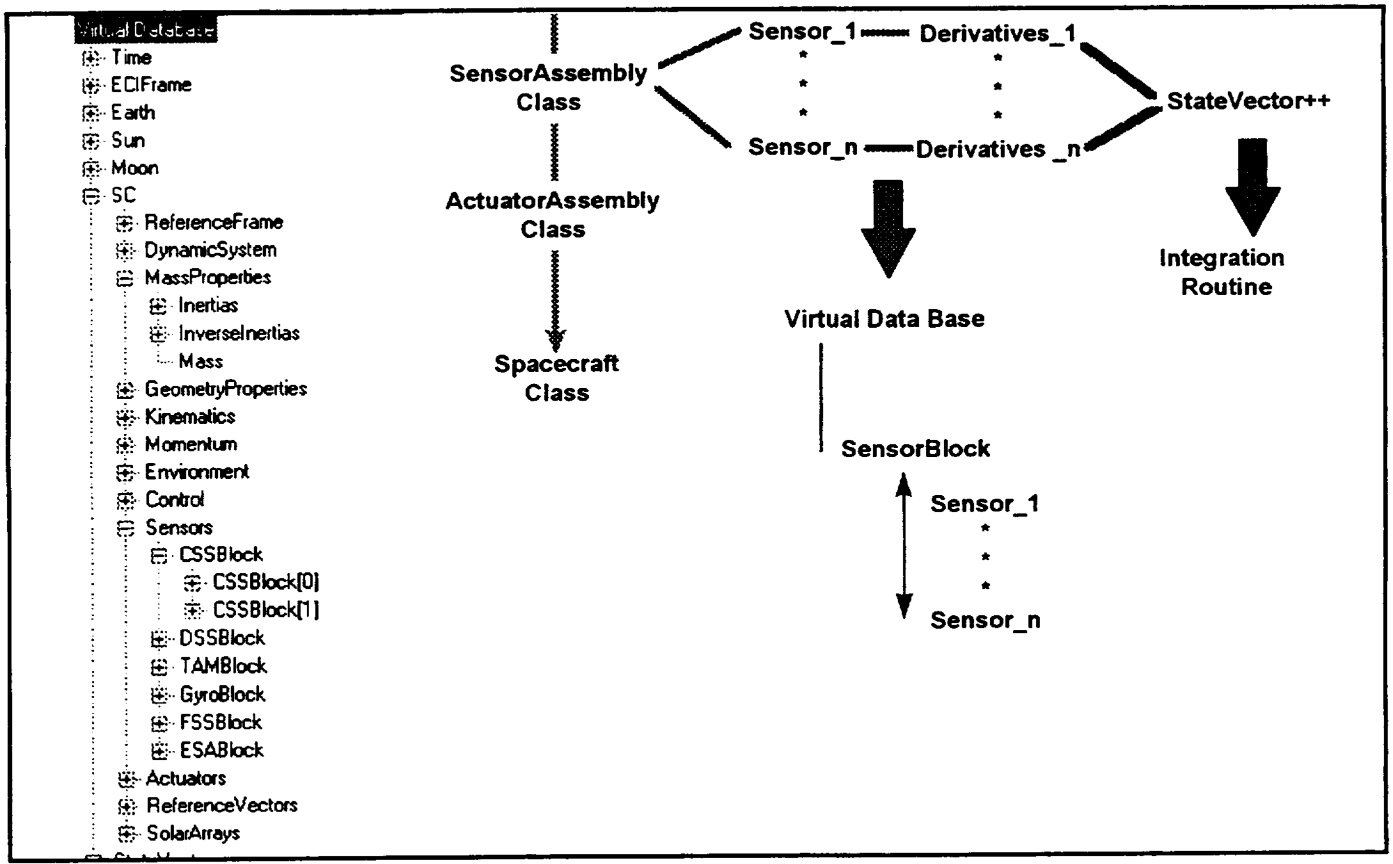


Figure 1. OOD Process Overview / Virtual DataBase

TRANSFORMATION ENGINE

A Transformation Engine, see Figure 3, was developed which enables the user to obtain the matrix transformations between any coordinate system in the simulation. The main features of this Xform Engine is that it uses relative coordinates to

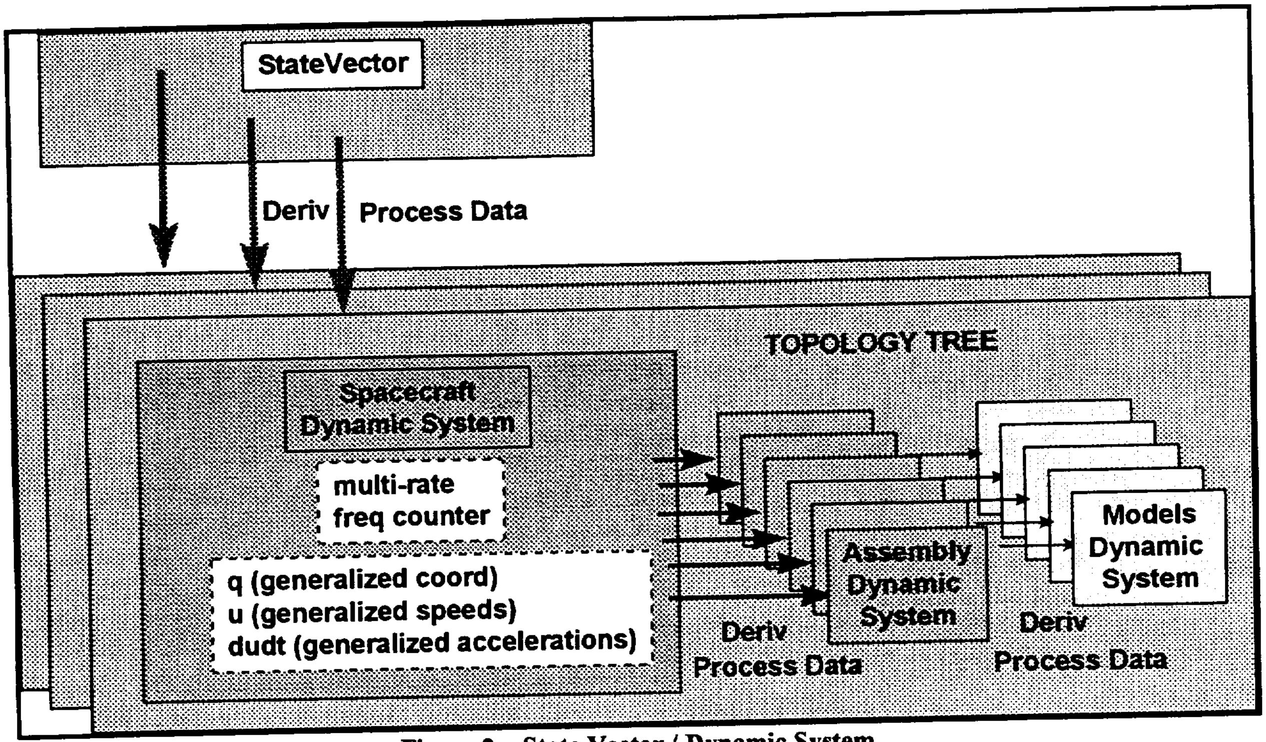
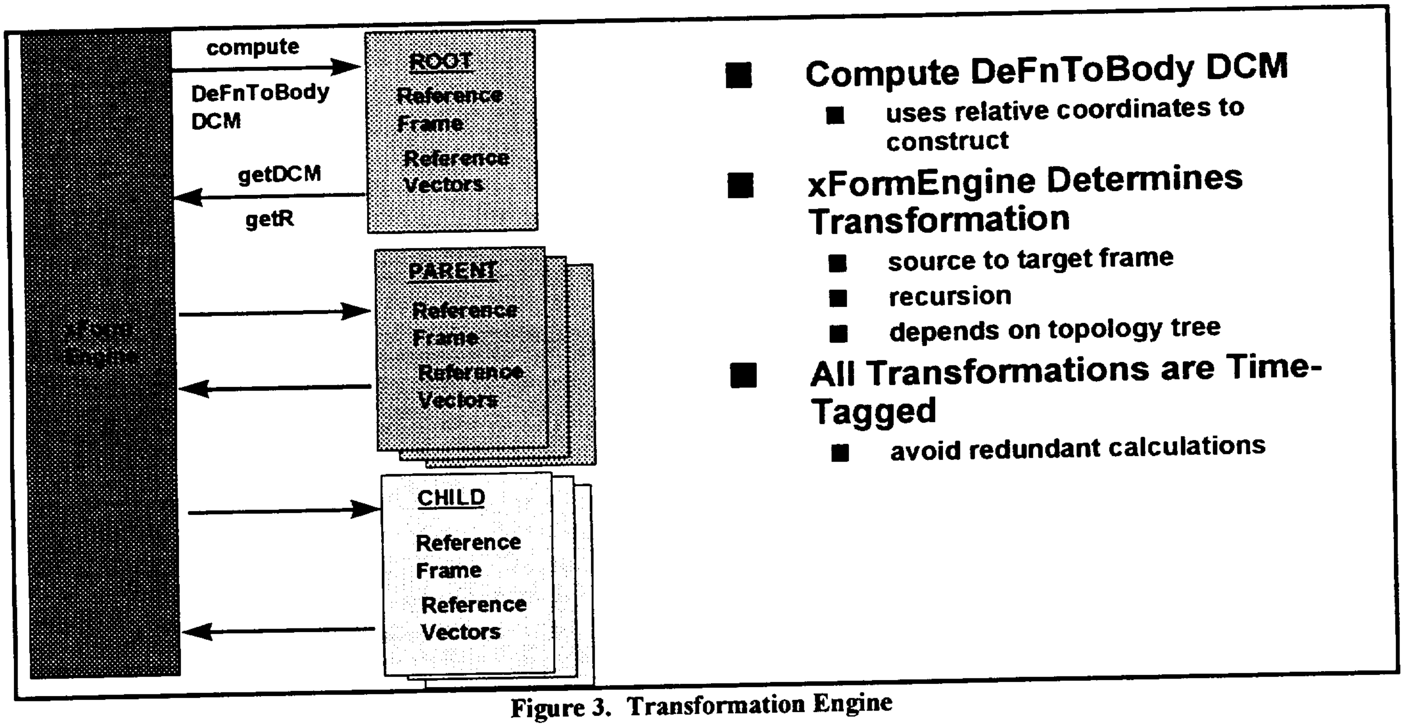


Figure 2. State Vector / Dynamic System



- Compute DeFnToBody DCM
  - uses relative coordinates to construct
- xFormEngine Determines Transformation
  - source to target frame
  - recursion
  - depends on topology tree
- All Transformations are Time-Tagged
  - avoid redundant calculations

Figure 3. Transformation Engine

construct a topological tree; a recursion algorithm is used to traverse the topological tree to determine the appropriate matrix transformation from the 'source' coordinate frame to the 'target' coordinate frame; time tagging is used to avoid redundant calculations. The Xform Engine has one root or reference frame (such as the Earth Centered Inertial, eci-frame) and many parent and child reference frames which lends itself to accommodating multiple rigid bodies. As the SensorAssemblyClasses or ActuatorAssemblyClasses are instantiated and added to the Virtual DataBase, the appropriate topology of the instantiated Class is autonomously added to the Xform Engine. This approach to tracking reference frames enables the user to view any variable or vector in any frame at any time, That is , the user can change coordinates frame views on the fly. For example, the User could view a coarse sun sensor on a rotating solar panel in its sensor frame or instantly view it in the spacecraft body or even the eci-frame.

VDBOBJECT, VIRTUALDATABASE, INTERPRETER

The VDBObject is the interface class between the user and the model, see Figure 4. Any class derived from VDBObject should provide virtual member functions Register(const char* typeDefn) and AttachToDatabase(const char* pathId). Register() submits those parameters which are to be made available to the user either for configuration (read-write) or display (read-only) and establishes the database structure as is clearly seen from the VDB Tree View. Only the user can access data directly via the front-end ,i.e., scripts, dialogs, etc. (Classes requiring data from other classes can only do so through member functions which enforce data protection and privileges. This keeps developers from having one model inadvertently contaminating the data of another model. Maintaining data integrity is particularly important as the simulation becomes larger and consequently more complex). AttachToDatabase() overlays the virtual database memory with the classes private data. Since the database memory exists at the lowest level of the run-time stack the data appears to be global to the user. This allows the user to access model parameters through scripts, drag and drop data onto plots, monitor windows, loggers, etc. by specifying the unique database path. As shown in Figure 5, the Interpreter parses scripts and generates highly efficient intermediate binary code as well as a symbol table. The backend then uses the symbol table to access data and execute the icode-instructions which operate on those data with almost the same performance as an executable. Finally, the VirtualDatabase class orchestrates all of the VDBObjects representing the models for saving and restoring, initializing, accessing and configuring.

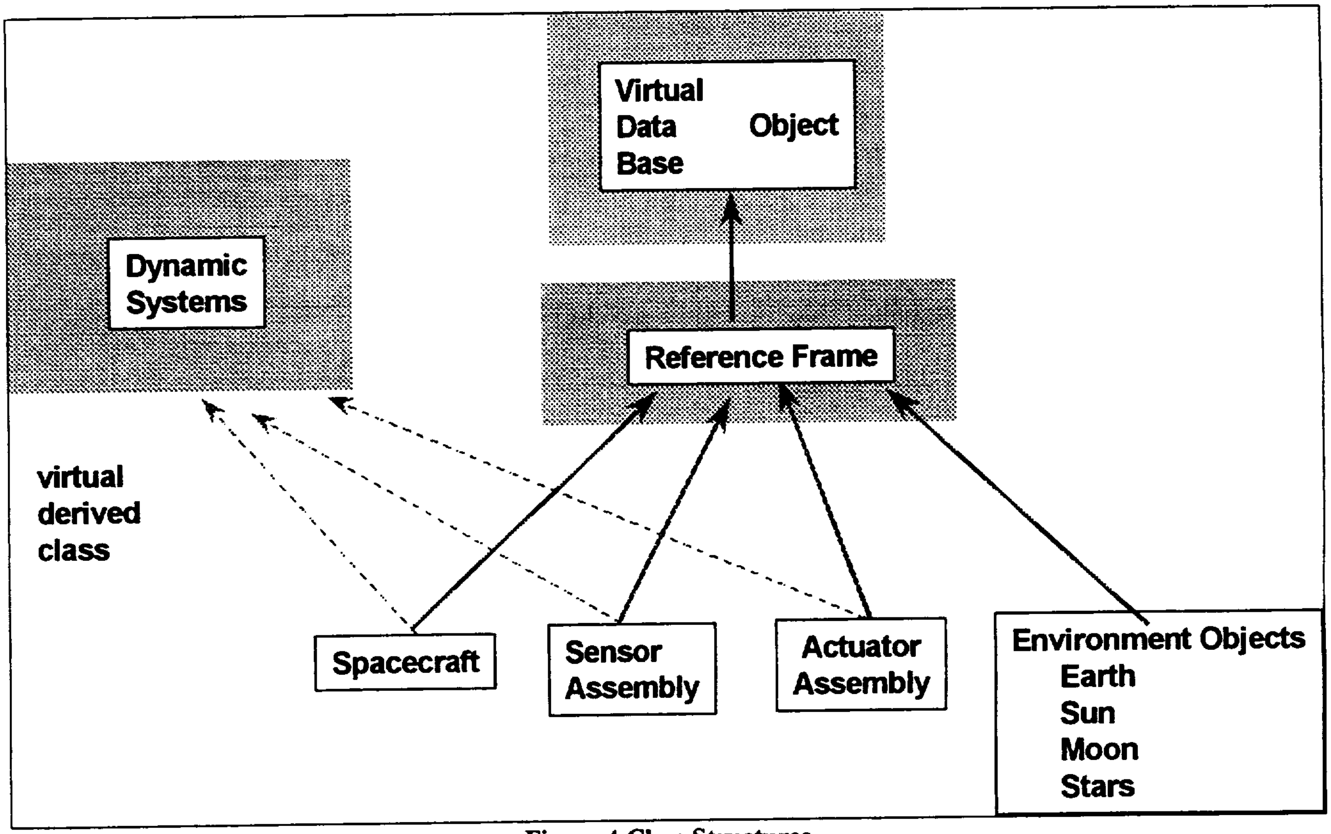


Figure 4 Class Structures

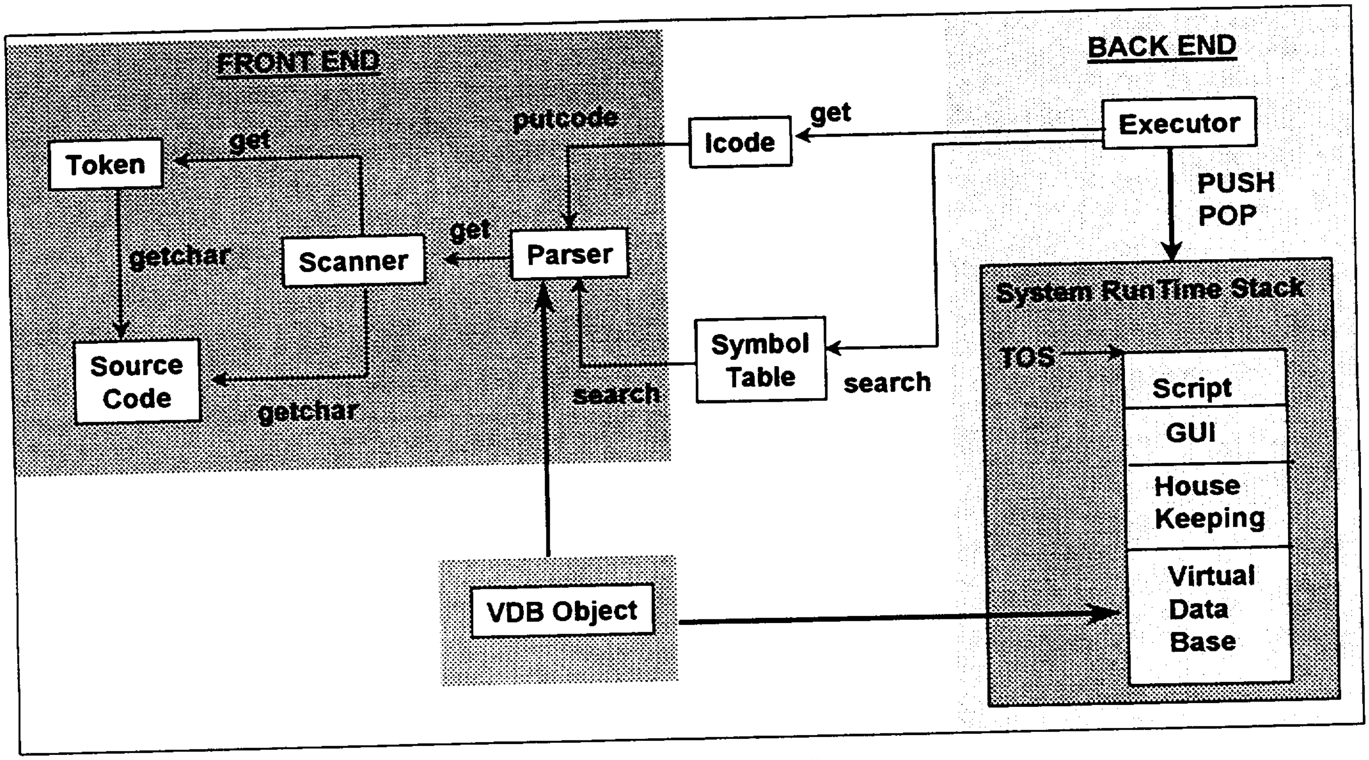


Figure 5 Virtual Data Base and Interpreter

MATRIX, VECTOR CLASSES

The base class Matrix, see upper left corner of Figure 6, provides the most generic operations and functions for mathematical matrix computations. For example, based on the dimensions of an instance of this class the inverse is calculated using a QR decomposition for a square matrix or a SVD decomposition for a matrix representing an overdetermined system. The classes Vector, Vector3x1 and Orthogonal3x3 override these operations and functions with code streamlined for performance. For example, Orthogonal3x3 overrides the inverse operation of the base class Matrix with a simple 3x3 matrix transpose. For mathematical objects such as quaternions, operations are overridden in order to implement quaternion algebra.

REFERENCE FRAME CLASSES

The base class ReferenceFrame class, see upper right corner of Figure 6, encapsulates all the kinematic quantities necessary for describing 6-DOF motion (position, velocity, acceleration, orientation, angular velocity, and angular acceleration. . The class is a pure abstract class in which any derived class must provide functions relating generalized coordinates, speeds and accelerations to their counterpart rectangular coordinates. These coordinate transformations allow a derived class to use generalized coordinates for internal use but must provide rectangular coordinates as a convention for interfacing with other frames. For orientation, an embedded base class "Orientation" allows the user to configure various parameterizations such as direction cosines, quaternion, or euler angle sequences, see lower left corner of Figure 6. In addition, the ReferenceFrame class contains links to the parent and children frames thus establishing the kinematics topology. The TranformationEngine, a friend class of ReferenceFrame, uses this private data and interface in order to develop the relative transformation from a given frame to any other frame by recursively traversing the topological path.

DYNAMIC SYSTEM, STATE VECTOR, TIME CLASSES

The dynamic system class is a pure abstract base class from which a derived class must override the member functions StateDerivatives(q,t) and ProcessData(t). This is because any class derived from DynamicSystem will autonomously be embedded within the total state vector by the friend class StateVector, see lower right corner of Figure 6. StateVector requires that such a function exist so that states can be integrated forward (or backward) in time. The user simply connects subsystems configuring the total system (e.g. ReactionWheelAssembly is composed of a desired number of individual reaction wheels). No other action is necessary since the StateVector class recursively traverses the dynamical tree calling each subsystem's StateDerivatives(q,t) function every minor cycle. At every major integration cycle, the StateVector class calls ProcessData(t). This provide an opportunity for the derived class to manipulate the generalized coordinates and speeds (for example, generating sensor data outputs or implementing a control law). Also embedded within the StateVector class is an instance of the Time class of which StateVector is the only friend class which has the privilege to increment the system clock. The time class provides commonly used parameterizations of time used by the models such as GMT, Julian Date, etc.

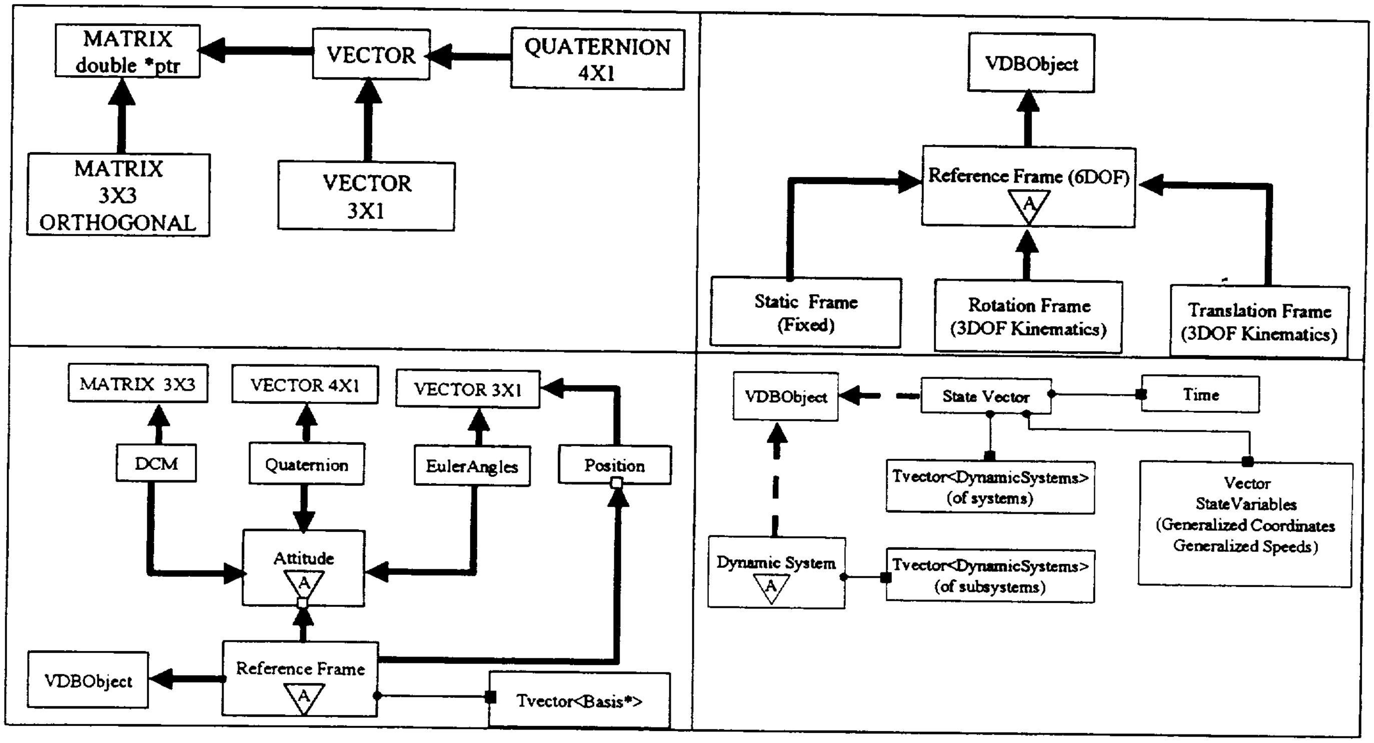


Figure 6 Base Classes

SENSOR AND SENSORASSEMBLY CLASSES

An example of a sensor class is the Gyroscope shown in the upper left corner of Figure 7. The Gyroscope class provides public interface member functions that return values such as analog and digital counts. That is, only the data that a physical gyroscope provides. There are many private member functions with implement various models composing the total gyroscope model such as misalignments, quantization, random walk and white noise, drift rates, dynamics, etc. The GyroAssembly class is a container class of Gyroscopes, derived from StaticFrame. This class also executes additional processing such as utilizing the TransformationEngine class to compute the inertial angular velocity of the frame to which it is mounted. Also the TransformationEngine resolves requests between SensorAssembly derived classes and those environment objects which are sensed (for example, the CoarseSunSensorAssembly makes a request to the Transformation Engine to determine the sun position within its own frame). It also collects the individual measurements produced by the component gyroscopes to be shipped back to the ServoMechanism class for processing.

ACTUATOR AND ACTUATORASSEMBLY CLASSES

An example of a sensor class is the ReactionWheel shown in the upper right corner of Figure 7. The ReactionWheel class provides public interface member functions that return values such as tachometer readings. That is, only the data that a physical reaction wheel provides. There are many private member functions with implement various models composing the total reaction wheel model such as misalignments, quantization, drag, the torque-speed curve, bearing noise, dynamics, etc. The ReactionWheelAssembly class is a container class of ReactionWheels, derived from the StaticFrame and MomentumStorage classes. This class also executes additional processing such as utilizing the TransformationEngine class to compute the total torque impressed on the frame to which it is mounted including gyroscopic disturbances. It also collects the individual torque commands produced by the ServoMechanism class to be shipped to each component ReactionWheel for processing. There are other implementations of ReactionWheelAssembly such as EnergyMomentumWheelAssembly and ControlMomentGyroAssembly.

JOINT AND JOINT ASSEMBLY CLASSES

The specialized classes StaticFrame, RotationFrame, and TranslationFrame override various virtual kinematics operations for optimizing commonly used reference frames in which the motion is constrained to being fixed, rotating, or translating, respectively, relative to its parent. An example of a joint class is shown in the lower left corner of Figure 7. For example, a frame which is fixed to its parent frame (such an a coarse sun sensor assembly mounted on a solar array) would not even attempt to compute the velocity, acceleration, angular velocity and angular acceleration relative to the solar array. Even more specialized classes (Revolute, Prismatic, Universal, etc.) are available for defining constraints for rigid bodies connected via joint configurations (TVector< Joint* >). The ReferenceFrame class and its derivations are essentially the "glue" which connect subsystems with each other and with their environment.

ENVIRONMENT CLASSES

An example of an environment class is the Earth shown in the lower right corner of Figure 7. The Earth class is derived from RotationFrame since it provides the transformation between the ECI and ECF reference frames and the pure abstract ExternalDisturbance class for which the virtual functions GetTorque() and GetForce must be overridden. Embedded within the Earth class are two instances of the SphericalHarmonics class (configurable order and degree) used for determining the gravitational field vector and the geomagnetic field vector. The SphericalHarmonics class accepts various normalizations (Schmidt, Gauss, WGS, Neumann). Also embedded is an atmospheric density model (Jacchia-Roberts) for computing aerodynamics force and torque impressed on a rigid body. The Earth class also provides public member constants commonly referenced such as polar and equatorial radii, mass, gravitational constant, etc.

RIGIDBODY, SERVOMECHANISM, CONTROLLER, PROCESS, SPACECRAFT CLASSES

Figure 8 presents the overall Spacecraft Class which is derived from the previously discussed Classes. The RigidBody class, derived from DynamicSystem and ReferenceFrame classes, has a vector of links to the ReferenceFrame base class. These links are other RigidBody classes representing appendages or JointAssembly classes representing joint constraints which link other RigidBody classes (e.g. SolarArray). Other parameters encapsulated within RigidBody are mass properties and geometric properties. The ServoMechanism class is derived from RigidBody and represents a controllable mechanical system. The ServoMechanism class is a container for the base classes SensorAssembly, ActuatorAssembly, and other subsystems. Also encapsulated within the class is the Controller class which is a container class of Processes. A Process class is a user-defined script which can be configured to execute at a desired frequency and priority. Much like the SensorAssembly and ActuatorAssembly classes the Controller (essentially a ProcessAssembly class) provides additional processing such as scheduling the individual Processes based on their frequency and priority. As a result, the ServoMechanism class is what defines the control system configuration. The Controller collects sensors data provided by the SensorAssembly classes then processes this data and determines torques to be issued to the ActuatorAssembly classes via user-defined scripts implementing a control law. Finally, an application of a ServoMechanism is the Spacecraft class which connects the ServoMechanism with an environment. For an earth-orbiter, the container of Environment models, TVector<ReferenceFrame*>, usually consists of instances of the Earth class, the Sun class, and the Moon class. Additionally, an environment can be configured with a number of instances of the Star class for Spacecraft configured with star sensors.

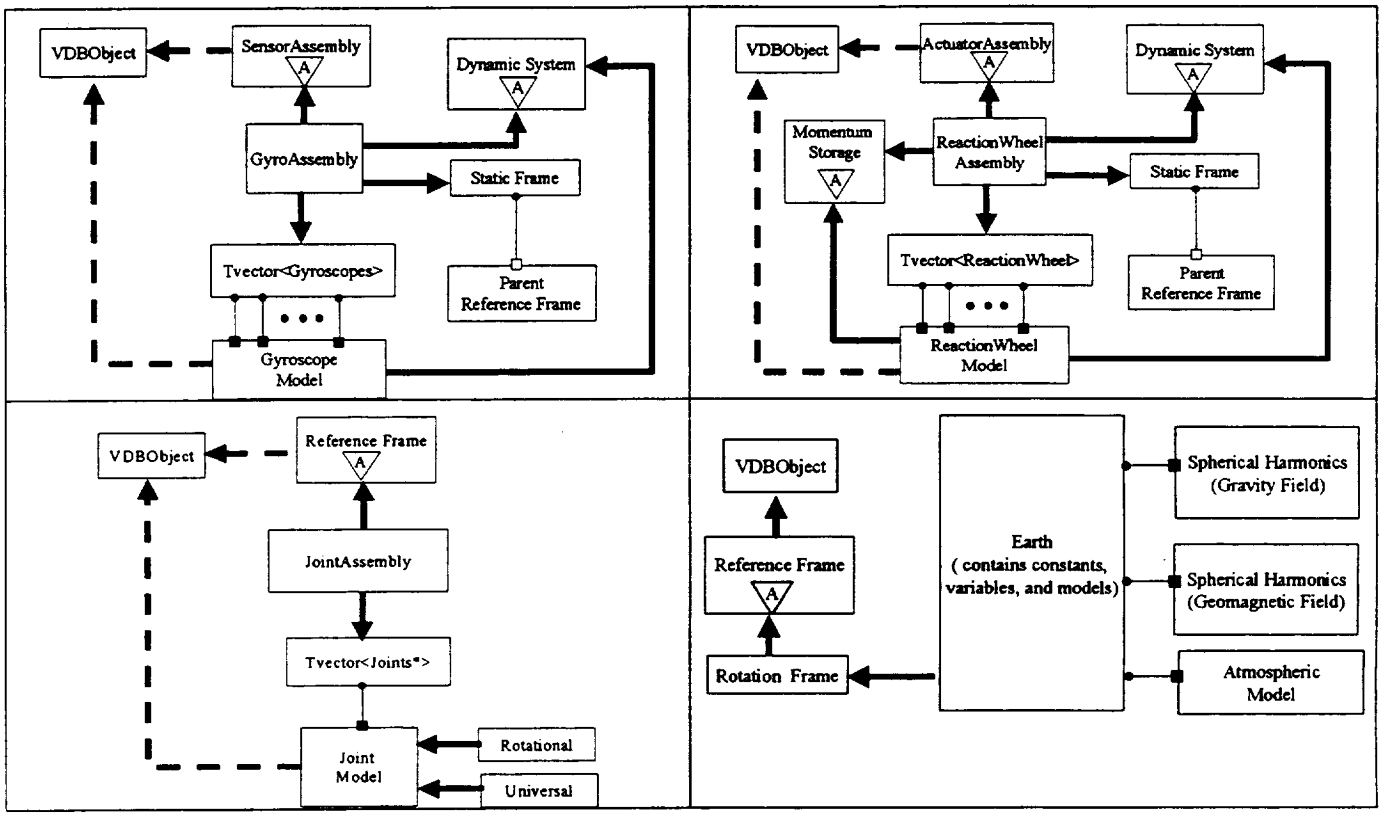


Figure 7 Example Of Derived Subsystem Classes

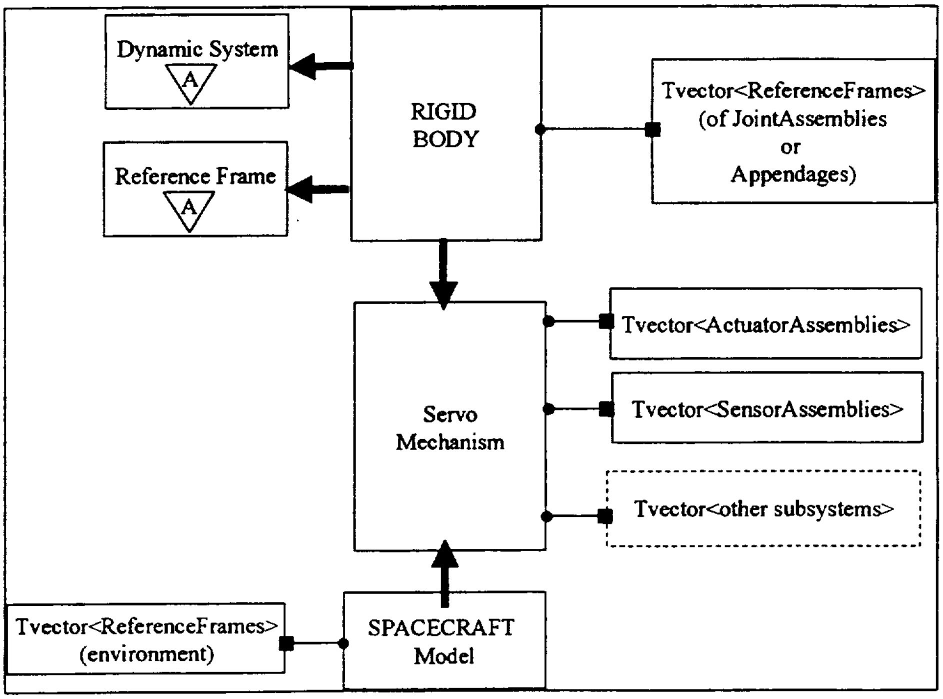

Dynamic System
A
Reference Frame
A
RIGID BODY
Tvector<ReferenceFrames> (of JointAssemblies or Appendages)
Servo Mechanism
Tvector<ActuatorAssemblies>
Tvector<SensorAssemblies>
Tvector<other subsystems>
Tvector<ReferenceFrames> (environment)
SPACECRAFT Model

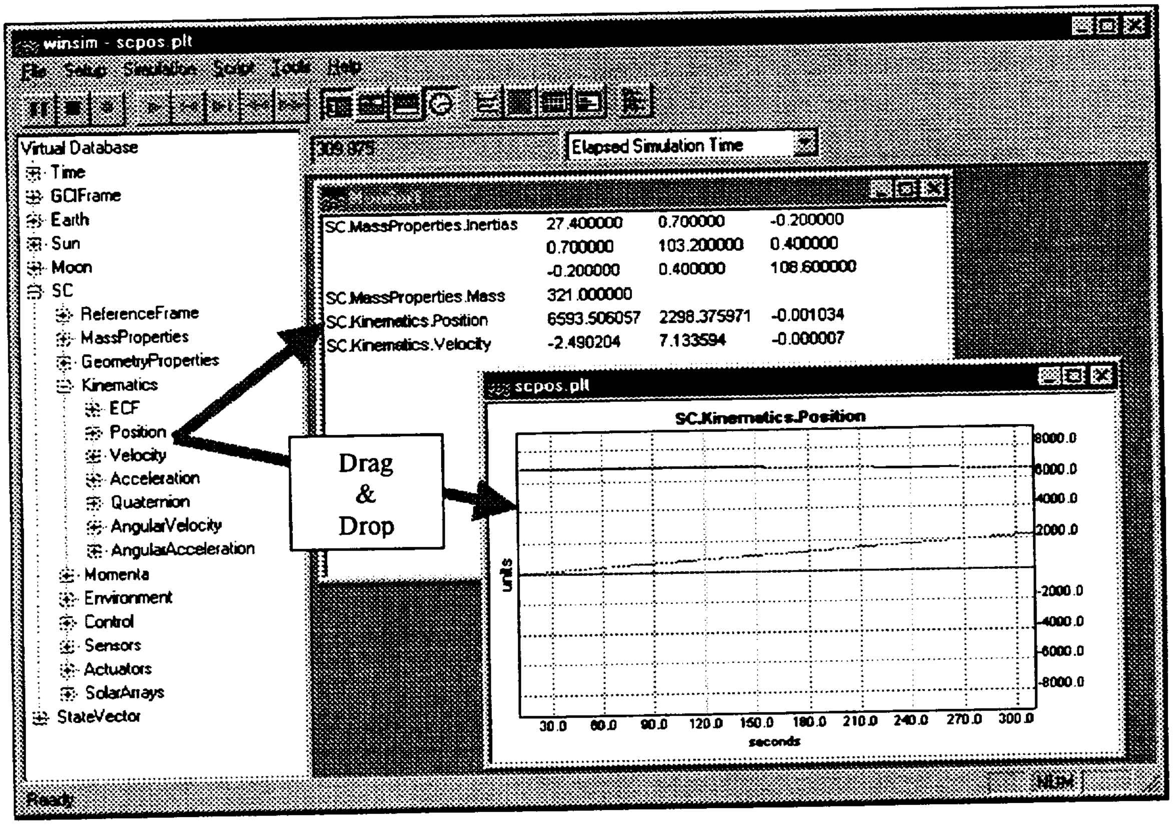
Figure 8 Spacecraft Class

IMPLEMENTATION OF OOD ARCHITECTURE

The OOD architecture has been implemented on a WindowsNT Operating System as shown in Figure 9. The Objected Oriented Windows environment enables the User to instantiate, within limits of the computer platform, as many plot, log or monitor windows as desired. Data is selected from the Virtual Data Base Structure and then "Drag & Drop" on the appropriate window for viewing. Figure 10 show the concept of how such a structure could be used for ADACS design, flight software verification and validation and support of ground operations. The scripting language functions like a 'STOL' testing language where it configures and controls execution of the simulation by enabling the user to modify parameters of the simulation on the fly.

FUTURE DIRECTIONS

The current Object Oriented Simulation Architecture for Rapid Spacecraft Prototyping has the capability to support the current spacecraft systems but requires changes at the development level such that each new spacecraft configuration would require changes to the code such as shown in the **INTRODUCTION**. A major feature of the current architecture is its extensibility to support the functional capability the ability to use this architecture for spacecraft prototyping with



```
SC.MassProperties.Inertias    27.400000     0.700000      -0.200000
                              0.700000      103.200000    0.400000
                              -0.200000     0.400000      108.600000
SC.MassProperties.Mass        321.000000
SC.Kinematics.Position        6593.506057   2298.375971   -0.001034
SC.Kinematics.Velocity        -2.490204     7.133594      -0.000007
```

Figure 9 Window Environment Using Object Oriented Architecture

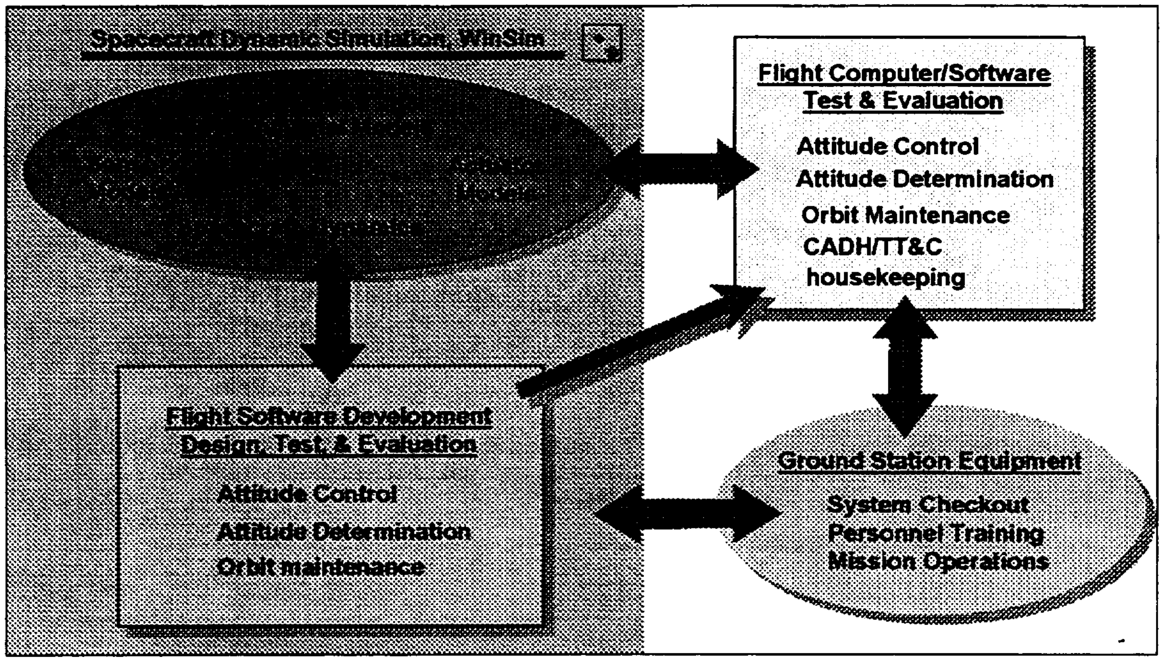


Figure 10 Overview of System Integration and Test

REFERENCES

1. "Scientific and Engineering C++, An Introduction with Advanced Techniques and Examples", John J. Barton, Lee R. Nackman, Addison-Wesley, Publishing Company, Inc. 1994

2. "The C Programming Language", B. Kernighan, D. Ritchie, Prentice-Hall,Inc., N.J. 1978

3. "Compiler Construction: Principles and Practice", K. Loudon, PWSPublishing Co.1997

CONTROL OF A THRUST ALIGNMENT TABLE FOR MODELING THE CONING DYNAMICS OF A SPINNING SPACECRAFT WITH A FOLLOWER FORCE*

Dominic Halsmer, J. Damon Bennett, Max DeHaven, and Vidar Ligard
Engineering and Physics Department
Oral Roberts University

ABSTRACT

This document presents a system controlling the motion of a spherical air bearing used in the modeling of spacecraft dynamics and controls in a laboratory environment. The system is part of the Spinning Rocket Simulator (SRS), used to simulate the coning of spacecraft during a thrusting stage. The reaction force at the spherical air bearing supporting the spacecraft model must coincide with the thrust axis of the model for proper simulation. Therefore, the bearing is translated in a circular path to introduce a centrifugal force. This horizontal force along with the gravitational reaction force at the bearing combines to simulate the direction of the spacecraft's thrust force. The control system receives attitude information from the spacecraft model via a laser beam embedded in the model that impinges on a photosensitive array. The non-linear system is controlled using high-speed lookup tables and digital techniques. A vector-controlled motor and a stepper motor are given the necessary signals to accurately control the turntable and platform supporting the air bearing. Preliminary performance data is presented. Mechanical elements of the table and platform are described in detail. A wireless (RF) data path for all devices on the spacecraft model to an off-table command computer is also described.

INTRODUCTION

The problem of coning instability in thrusting, spinning bodies with internal mass motion has been made evident by spacecraft flight anomalies during orbital injection maneuvers (ref. 1). During these maneuvers, the thrust axis coincides with the nominal spin axis of the spacecraft. However, some vehicles experienced dangerous levels of coning toward the end of the burn. A possible mechanism for this instability is the interaction between the nutation mode of the vehicle and the motion of propellant slag trapped in the annular region around the submerged nozzle exit cone. The development and analysis of mathematical models that capture the coupled-oscillatory nature of this problem have shed light on the dynamics, and suggested novel control approaches (ref. 2). A passive mass-spring-damper control device termed a Passive Coning Attenuator (PCA) has been proposed to eliminate/suppress coning during this type of maneuver. Necessary and sufficient conditions for asymptotic stability of coning motion and expressions for optimal design parameters have been derived in terms of general system parameters. Peak performance in terms of the time constant of coning attenuation has also been quantified (ref. 3).

However, there are significant limitations inherent to the analytical approach. It is difficult to accurately represent the behavior and interaction of more complex types of mass motion such as that of fluids. Furthermore, analysis and computer simulation does not always represent the actual characteristics of the necessary hardware elements. For these reasons, and also to validate the proposed PCA control device, an experimental approach is desirable. A spherical air bearing can be used to allow virtually frictionless three-dimensional rotational motion of a spinning spacecraft model. Until recently however, it was not known how to model the dynamics of a spinning *thrusting* spacecraft in a ground-based laboratory. A solution to this problem was first published by Dr. Rudy Meyer (ref. 4). It involves the acceleration of a gas bearing along a near circular path based on model attitude information. This allows the addition of a controlled amount of centrifugal force. In this way the resultant reaction force on the spacecraft model remains aligned at all times with the simulated thrust axis. The dynamics and control of such an

*This work is funded by a grant from NASA Goddard Space Flight Center.

apparatus, termed the Spinning Rocket Simulator (SRS), have been analyzed and presented, including accuracy and performance predictions for a specific example (ref. 5). This type of apparatus is currently being developed by an interdisciplinary team of undergraduate engineering and science students and faculty at Oral Roberts University with the help of Goddard Space Flight Center. The wealth of educational benefits resulting from this project has also been presented (ref. 6).

AIR BEARING AND SPACECRAFT MODEL

The heart of the apparatus is the spherical air bearing. This was also the first component to be developed. Its radius of curvature is 6.35 cm (2.5 in) and allows 360° of rotation about a vertical axis and up to 15° of rotation about transverse axes. The air passages were recently enlarged based on analysis of flow provided by a two-stage air compressor. Tests were conducted to verify that a tank pressure of 1.2 MPa (175 psig) adequately floats a spacecraft model weighing 667 N (150 lb) at a flow rate of approximately 0.4 m^3/min (14 cfm). The spacecraft model is designed to be easily adjusted to duplicate the inertia ratio of a wide range of prolate or oblate configurations. The model contains a horizontal plate for mounting sensors and other equipment, and remotely-controlled movable masses for balancing purposes. Mounted on the spacecraft model are four small, remotely-controlled, tangentially-oriented fans which provide a total of .03 Nm (.022 ft-lb) of spin torque. Spin tests indicate that these fans will maintain a spin speed of approximately 60 rpm in the presence of atmospheric drag on the model.

THRUST ALIGNMENT TABLE

Introduction

The testing of the Spinning Rocket Simulator requires that the simulated thrust force be aligned with the nominal spin axis of the spacecraft. While the cone angle is zero the thrust force direction is accurately simulated by the existing gravitational reaction force (Figure 1), but as the model begins to cone, the simulated thrust force is no longer vertical.

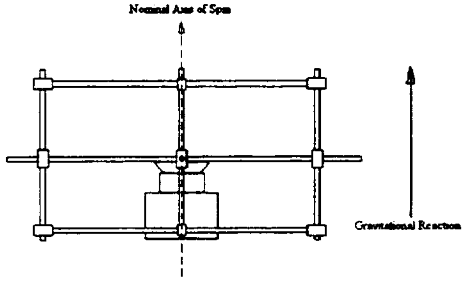


Figure 1. Zero Cone Angle

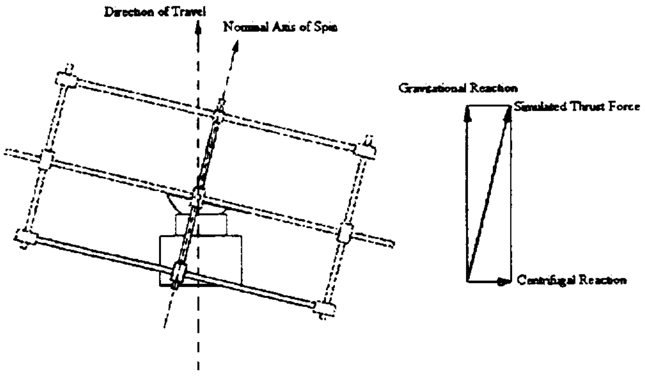


Figure 2. 14° Cone Angle

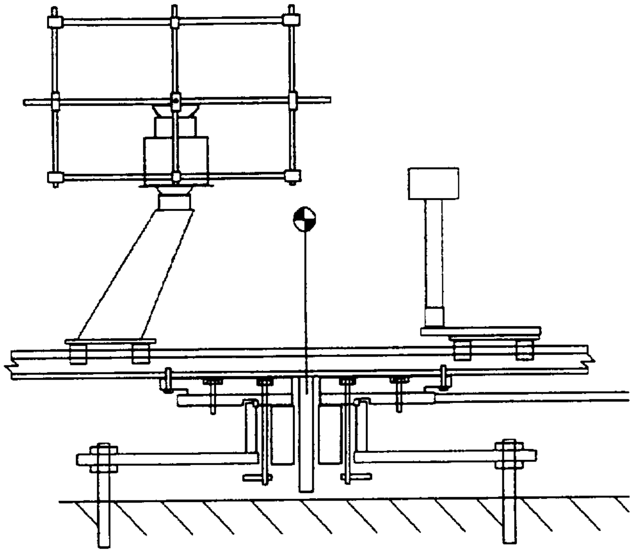


Figure 3. Turn-Table

When the model begins to cone (Figure 2), small amounts of centrifugal acceleration must be added to the existing gravitational acceleration so that the simulated thrusting force remains aligned with the model's thrust axis. For this purpose, a turntable has been build to keep the thrust force aligned with the nominal spin axis. The turntable by necessity has two degrees of freedom: the radius of spin (R, measured from the center of the turntable to the vertical axis that passes through the center of the spherical air bearing, Figure 4) and the angular motion (RPM, referring to the turntable's rate of spin, Figure 5). The turntable angular velocity, for the purposes of simulation, must be constrained to the precession rate of the model. As the model begins to cone and precess, the radius will be varied to keep the angle of the resultant simulated thrust force aligned with the thrust axis.

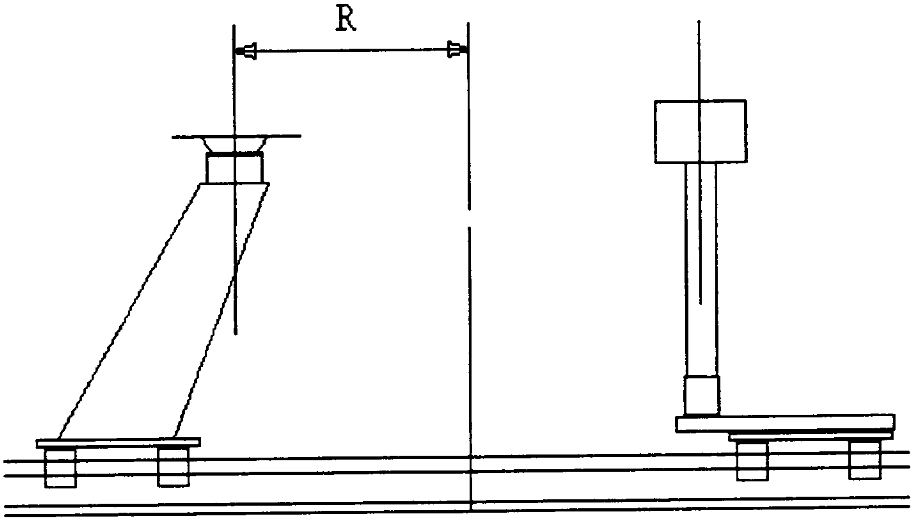


Figure 4. Radius of Spin

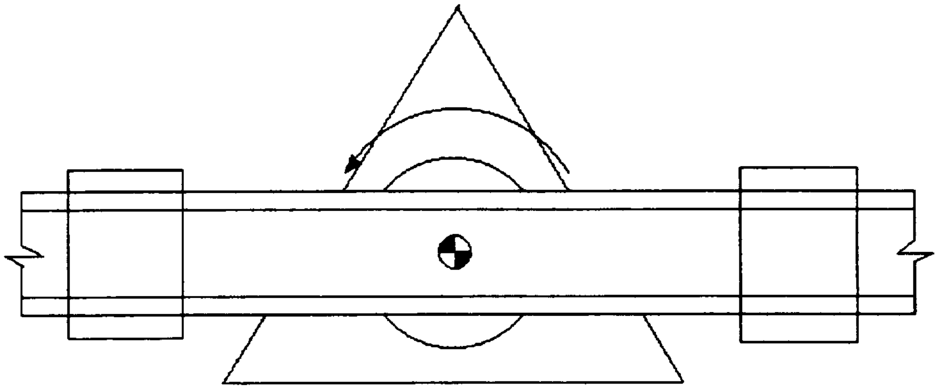


FIGURE 5 Angular Velocity

Specifications

The thrust alignment table must meet the following specifications:

1. spin so as to match the precession rate of the model
2. allow the spherical air bearing to move radially from the center of the table to the limit of motion ≅1.22m (48 in)
3. balanced (statically and dynamically)
4. support the weight of the model while bending less than 0.5° under static loading and 1° under dynamic loading
5. supply the spherical air bearing with compressed air
6. supply the onboard systems with a pathway for signal and power

System Overview

Lower Systems (Figure 6)

The interface plate is the central piece that connects the lower system to the upper system. The upper surface of the interface plate is bolted to the linear track. The lower surface of the interface plate will be bolted to the driven sprocket. The driven sprocket has been machined to mate (interference fit) with the inner race of the thrust bearing. The outer race of the thrust bearing mates (press fit) with and is supported by the thrust bearing column, a short hollow spacer used to lift the linear track to prevent interference with the table base plate bolts. The thrust bearing column is mounted around the hole in the center of the table base plate. The table base plate is triangular to allow for leveling.

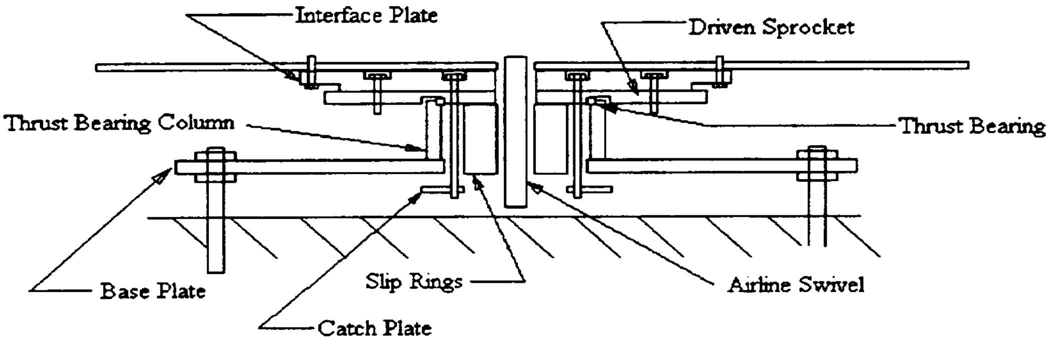


Figure 6. Lower & Inner Systems

The last piece is the catch plate. The catch plate is a failure mode design that will keep the upper system from departing the simulator in the event that the turntable becomes unbalanced. The catch plate is mounted on extension bolts that are fastened to the interface plate.

Inner System (Figure 6)

The spherical air bearing must have compressed air and the onboard control systems must have power. To supply air to the spherical air bearing a 1.9cm (¾ in) air-line swivel will be run up through the center of the turntable. The onboard control systems need to have small amounts of power and must be able to send signals out for analysis and recording. To this end a set of through bore sliprings are used since the center of the turntable is taken by the air-line swivel. There are twelve signal rings in the set. Since the power requirements for the onboard control systems are less than 250V/5A the signal rings will be more than adequate to supply power as well as pass signal.

Upper System (Figure 7)

The linear track is mounted on the linear track base plate. Two carts mounted on opposing ends of the track are connected by a twin lead ball screw so that they can be driven equally along the radial path. On one of the carts is the spherical air bearing column. This holds the spherical air bearing above the track, high enough that there is no interference between the model and the track.

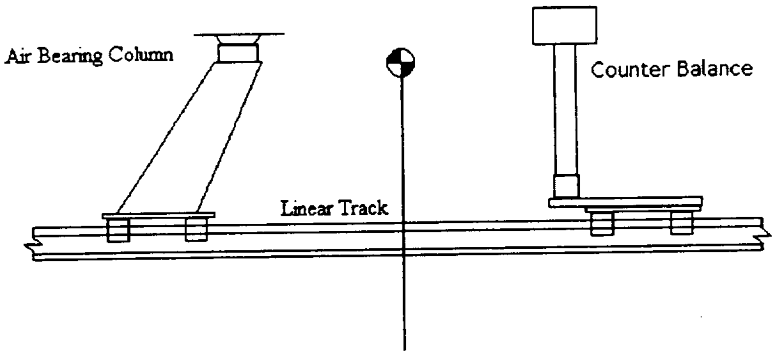


Figure 7. Turntable

On the cart opposing the Spherical Air Bearing is a counter-balance that is capable of dynamic as well as static balancing. Statically balancing the model would only require the same amount of mass be placed equidistant from the center of the turntable. Since the system will not be operating under static conditions, it must also be dynamically balanced. Dynamically balancing the system requires that the center of mass of the counter balance be at the same height and distance from the center of the turntable as the center of mass of the spherical air bearing system. To this end, the counter-balance is divided into two equal halves and then placed on a frame (Figure 8) that resembles a foot ball goal post. The frame places the center of mass of the counter balance system at the same height as the spherical air bearing system and allows for the linear drive system to bring both to the center position with no interference.

Physical Limitations

The system has the following limitations:

1. turntable maximum angular velocity limitation of 30 RPM
2. radial maximum velocity limit of 7.6 cm/sec (3 in/sec)
3. twelve rings for signal/power, each power circuit = 2 rings
4. Limited spacecraft configurations (inertia ratios) can be modeled.
   A study is being conducted on the configuration range that can be modeled with the system.

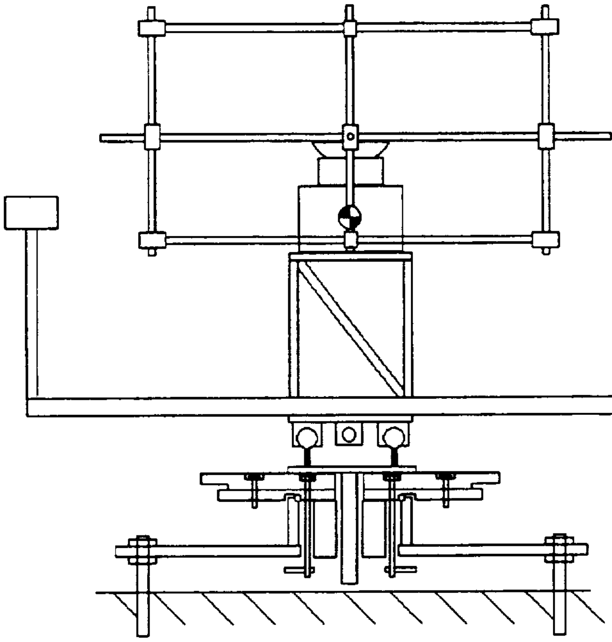


Figure 8. Turntable (End View)

OVERVIEW OF TABLE CONTROL SYSTEM

The purpose of the control system described in this section is to accurately control the bearing platform and turntable using the output of the optical attitude sensing system. A photosensitive array receives a laser beam originating at the center of rotation of the model. This optical sensing array provides Cartesian coordinates of the on-model laser beam, making the dynamic relationships highly non-linear. Consequently, the input signals are digitized and converted using high-performance digital lookup tables. RISC microcontrollers operate a stepper motor which drives the model along the table radius, and an a.c. motor provides the angular motion. The mechanics of the turntable provides the feedback necessary for closed loop control.

DEFINITIONS

Cone Angle, θ -- The angle between the angular momentum vector and the thrust axis (nominal spin axis) of the spacecraft is known as the cone angle, or nutation angle. The SRS simulates the direction of the angular momentum vector as vertical when the cone angle is zero. Thus the cone angle is the angle between vertical and the thrust axis (typically the minor axis) of the model.

Precession Angle, ϕ -- The thrust axis of the model projected onto a horizontal surface gives a line. The angle between this line and another fixed horizontal line is known as the precession angle. The precession angle along with the cone angle completely describes the direction of the thrust axis at any instant in time.

Turntable Angle, Ω and AC Vector Controlled Motor -- The turntable angle is simply the rotational angle of the table. The motor associated with this angular motion is known as the main drive motor, or the vector controlled a.c. motor.

Error Angle, $\varepsilon = \phi - \Omega$ -- The bearing and optical sensing plate rotate along with the table. Therefore, the "precession angle" from the sensing array is really the error angle, which ideally is always zero.

Radius, R and Stepper Motor -- The radius is the horizontal distance between the center of the turntable and the center of the bearing. A stepper motor driving a screw controls this radius.

X and Y -- Cartesian coordinates of the laser beam endpoint on the sensing plate.

Height, h -- This is the shortest distance from the center of rotation of the model to the PSD.

Gravitational Acceleration, g -- This is acceleration due to gravity on earth.

THRUST ALIGNMENT CONTROLLER

The thrust alignment controller ensures accurate motion of the turntable's two degrees of freedom, namely, the radius (R) and the turntable angle (Ω). A laser beam originating at the center of rotation of the model illuminates a photosensitive array (PSD). This provides the necessary attitude data for controlling the table. The design utilizes high-speed digital techniques, allowing for high flexibility while providing the necessary output data for test analysis. Figure 9 below shows a conceptual side view and top view of the simulator with pertinent control variables.

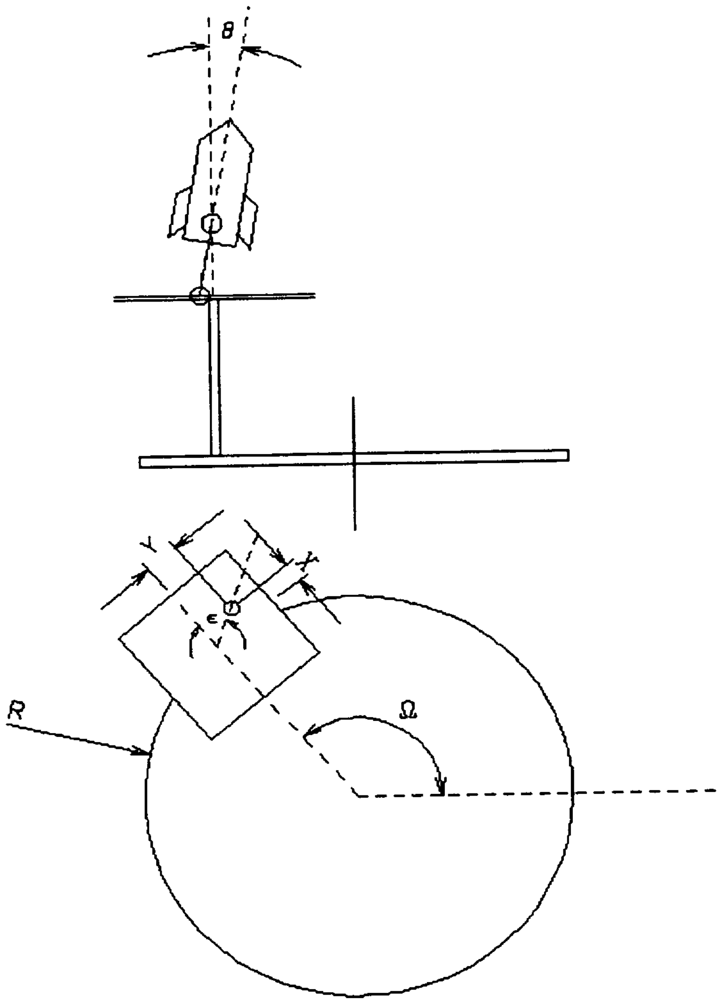


Figure 9. Table Control Variables

The side view illustrates the center of rotation of the model and the laser beam termination point (marked by black dots), and the cone angle, θ. On the top view of the simulator, the black dot marks the end of the laser beam on the PSD plate. Since the PSD rotates along with the table, the angle (ε) shown is the error angle, according to equation (3). The attitude sensing array provides the laser beam's terminating point in Cartesian coordinates. Consequently, the cone angle and precession angles are given by non-linear equations (1) and (2) below.

$$\Theta=\arctan\left(\frac{\sqrt{X^2+Y^2}}{h}\right) \qquad (1)$$

$$\varepsilon=\arctan\left(\frac{Y}{X}\right) \qquad (2)$$

$$\phi-\Omega=\varepsilon \qquad (3)$$

Design Requirements

To introduce the centrifugal acceleration necessary to align the simulated thrust axis with the angular momentum vector, dynamics show the following must be true (ref. 5).

$$R=\frac{g\tan(\Theta)}{\dot{\phi}^2} \qquad (4)$$

$$\Omega=\phi \qquad (5)$$

Design Solution

A digital controller is utilized for the system. This provides the desired flexibility and enables interface with another computer analyzing test data. Arrays of digital EPROM lookup tables handle the non-linear equations (1), (2), and (4) in a high-performance environment, introducing delays of only 100 ns. Signal flow and control algorithms are implemented by microcontrollers.

Desired for its position accuracy without using feedback, a stepper motor is used for the radial drive. To a step input, this motor is ramped to its maximum speed, and decelerated at its destination. A vector controlled AC motor provides high torque at low speed for the main drive. Position feedback is provided by an encoder on its shaft giving 4096 pulses per revolution. An integrating control algorithm is used for the main drive to eliminate steady state error to a ramp input under a wide range of inertia. The control system block diagram in the s-domain is shown in Figure 10.

Testing and Results

All stages in the controller have been tested and verified numerically. To a step command, the radius accelerates and moves to its position where it decelerates, as designed. Sending the motor to its home position after an extended random input shows that no steps are skipped or missed, verifying that feedback is not necessary for this drive. The motor's maximum speed translates to 4in/sec. The main drive, likewise, is controllable according to specifications. The main drive settles to a ramp input in less than one second under no load inertia.

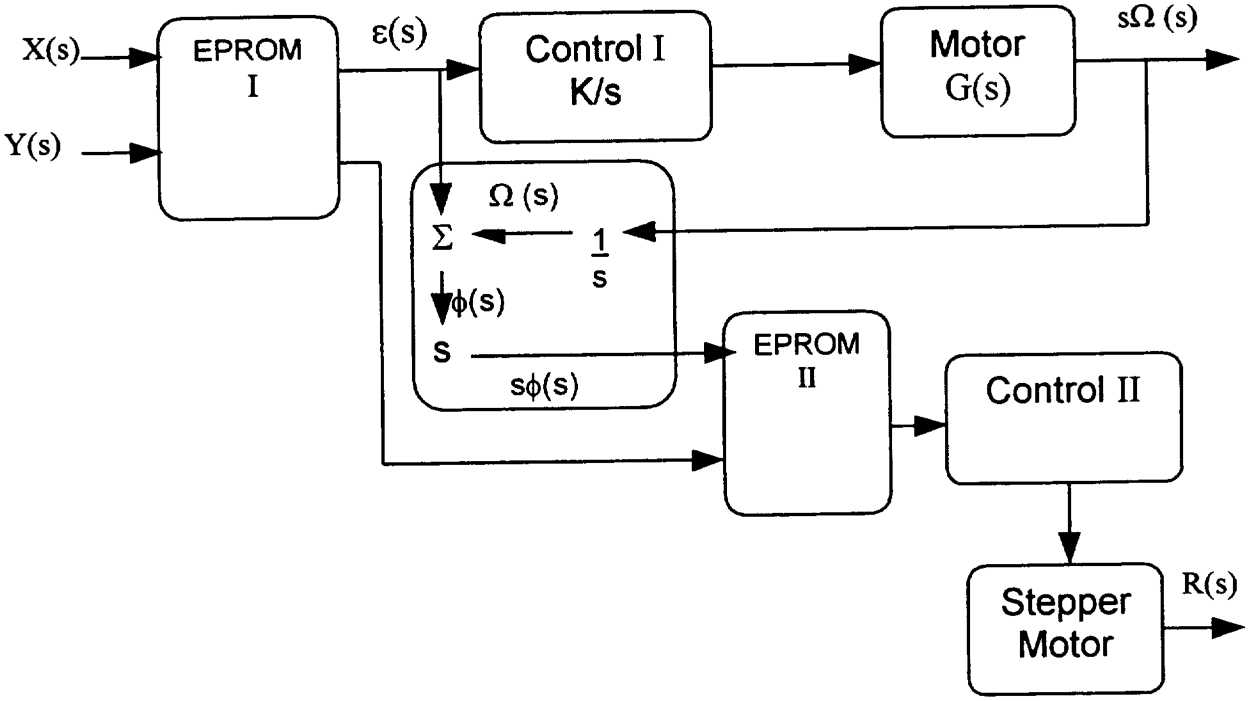


Figure 10. Control System Block Diagram

WIRELESS DATA PATHWAY

Introduction

The wireless contact project is a necessary sub-system of the spinning rocket simulator project. Its purpose is to provide a wireless data path for all the devices on the spinning spacecraft model to a command computer away from the spacecraft model. This project will allow the testing team to send information to the wireless contact project and receive it from the other end for the purpose of controlling the spacecraft model and acquiring data from the model. The necessity of a wireless link from the command computer to the model is due to the six degrees of freedom needed by the model for simulating a true spinning spacecraft under thrust.

Project Description

This project utilizes state-of-the-art microcontrollers throughout its design to achieve a maximum of flexibility for future testing, adaptability, and advanced processing of information. The total hardware design makes use of 30 microcontrollers allowing for 16 analog inputs from devices on the model and a maximum throughput of 400,000-baud full duplex. It also features four stepper motor controllers for the stepper motors on board the spacecraft model, two pulse-width-modulation drivers for dc motor control, and a regulated power input to the data processing devices to make use of an unreliable battery powered input.

The project is broken up into two basic platforms: the section mounted onto the spacecraft model and the section that connects to the command computer. The project is further broken down by the wireless functions it performs. Thus there are two interconnected sections on board the spacecraft model, the data reception pc-board and the data transmission pc-board. There are also two interconnected sections by the command computer, the data receiving pc-board and the data transmitting pc-board. The following subsections will describe the design content in each of these four components.

The On-Board Data Acquisition System

The data acquisition subsystem (Figure 11) is made up of eight A/D PIC16C715 microcontrollers, which may each, receive two analog inputs with reference to the regulated supply voltage.

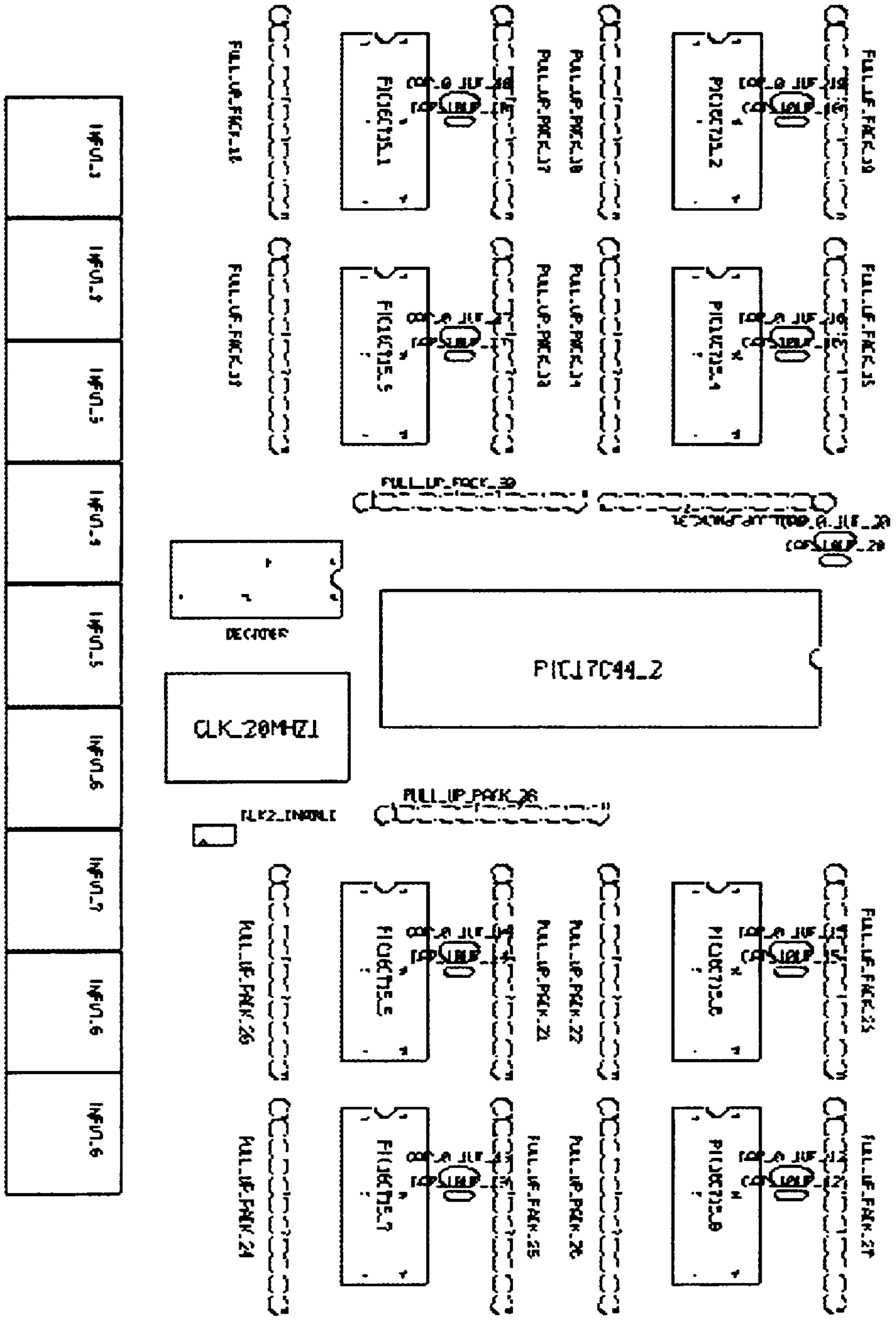


Figure 11. Data Acquisition Subsystem

The eight microcontrollers share a common eight-line output bus, which is also connected to the input master microcontroller. The input master microcontroller is a PIC17C44 that has USART capability. It addresses each of the A/D controllers using a 74LS138 demultiplexer after pulsing a data latch line. This forms a "freeze frame" picture of what is occurring on the spacecraft model. Following data capture the master microcontroller passes the information synchronously to the on-board transmission subsystem.

The On-Board Data Transmission System

The on-board data transmission system (Fig. 12) is made up of five microcontrollers and four 900 MHz high-speed transmitters. Its purpose is to transmit the data collected from the devices on board to the command side receivers, which will deliver the data to the computer for analysis. Each of the four transmitters is modulated asynchronously by a PIC16C63A microcontroller. The transmitting microcontrollers are provided with transmission data from a PIC17C44 master microcontroller, which places the data a common bus and negotiates the data acknowledgement through two control lines to each of the four transmitting microcontrollers. The master microcontroller receives its information from the master controller in the data acquisition/input stage of the on-board subsystem.

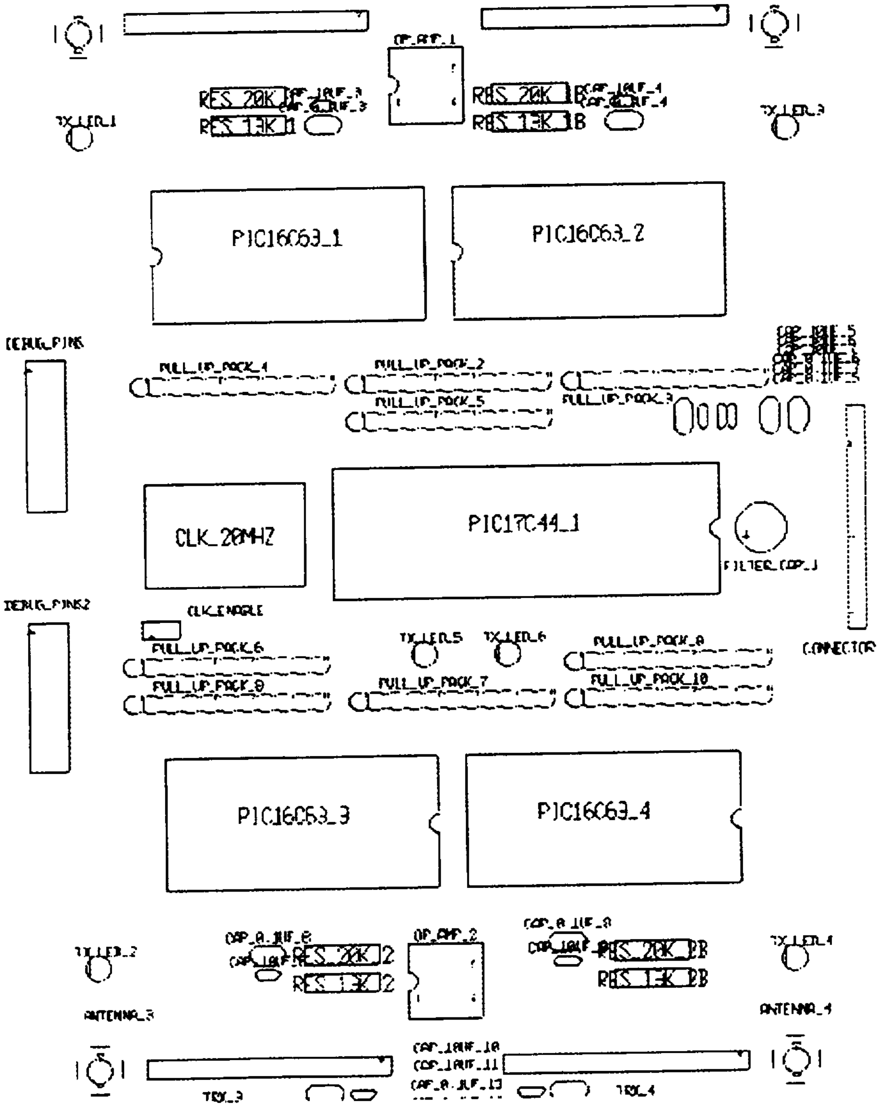


Figure 12. On-Board Transmitter

The On-Board Data Reception System

The on-board data reception system (Fig. 13) is comprised of 5 microcontrollers and four 900 MHz high-speed receivers. The purpose of the system is to receive the incoming computer data from the command-side transmission subsystem. It then synchronously sends the command information to the output stage controller. The four receivers capture the transmitted data and asynchronously pass the data to the four receiving PIC16C63A microcontrollers. The receiving microcontrollers then cue the output master controller, and, under the direction of the receiving master microcontroller, pass the data through the connected bus to the master receiver microcontroller.

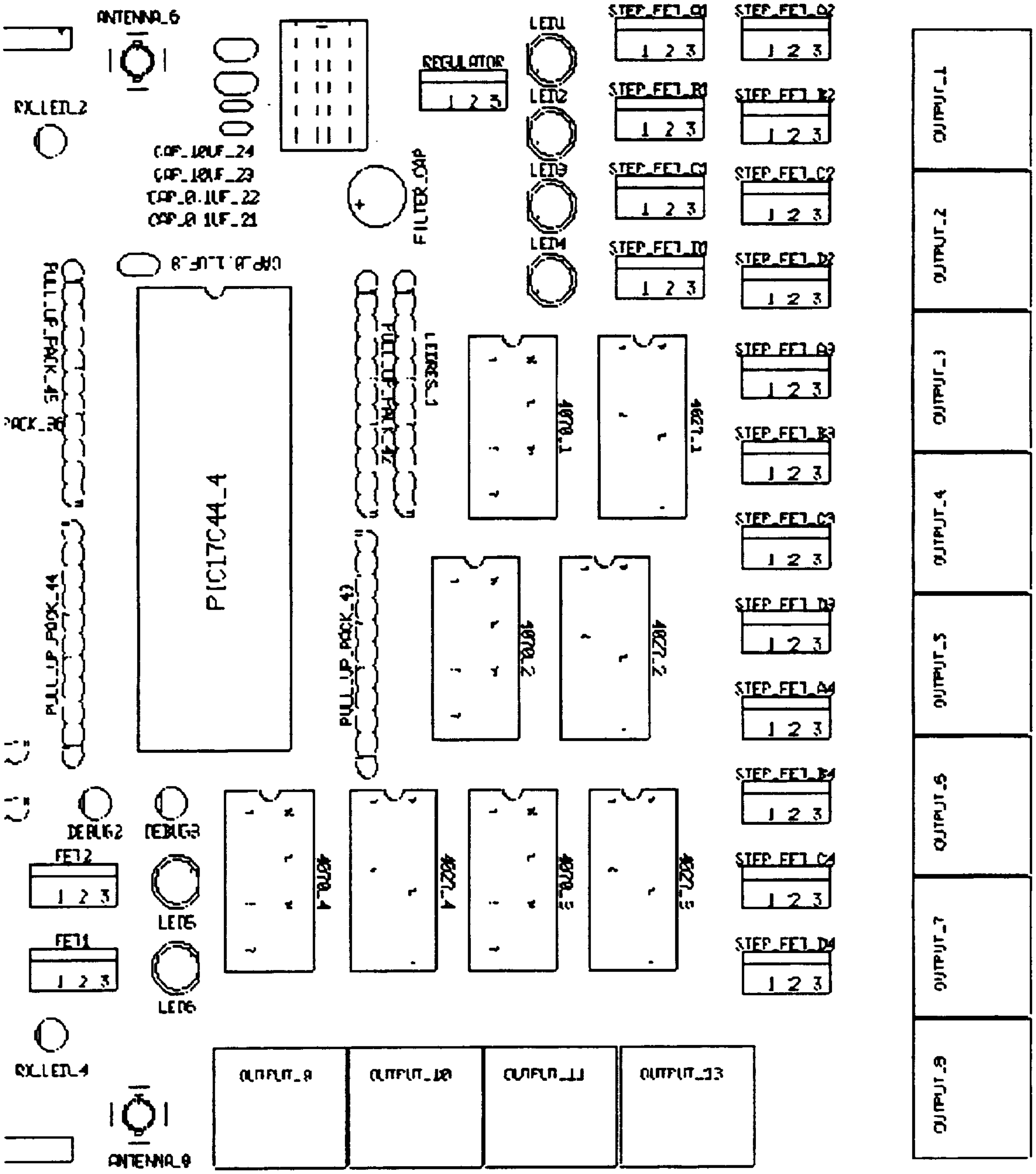


Figure 13. On-Board Receiver

The Controls System

The on-board controls system (Fig. 14) consists of a PIC17C44 acting as the master output controller. This microcontroller handles all the output functions on the spacecraft model.

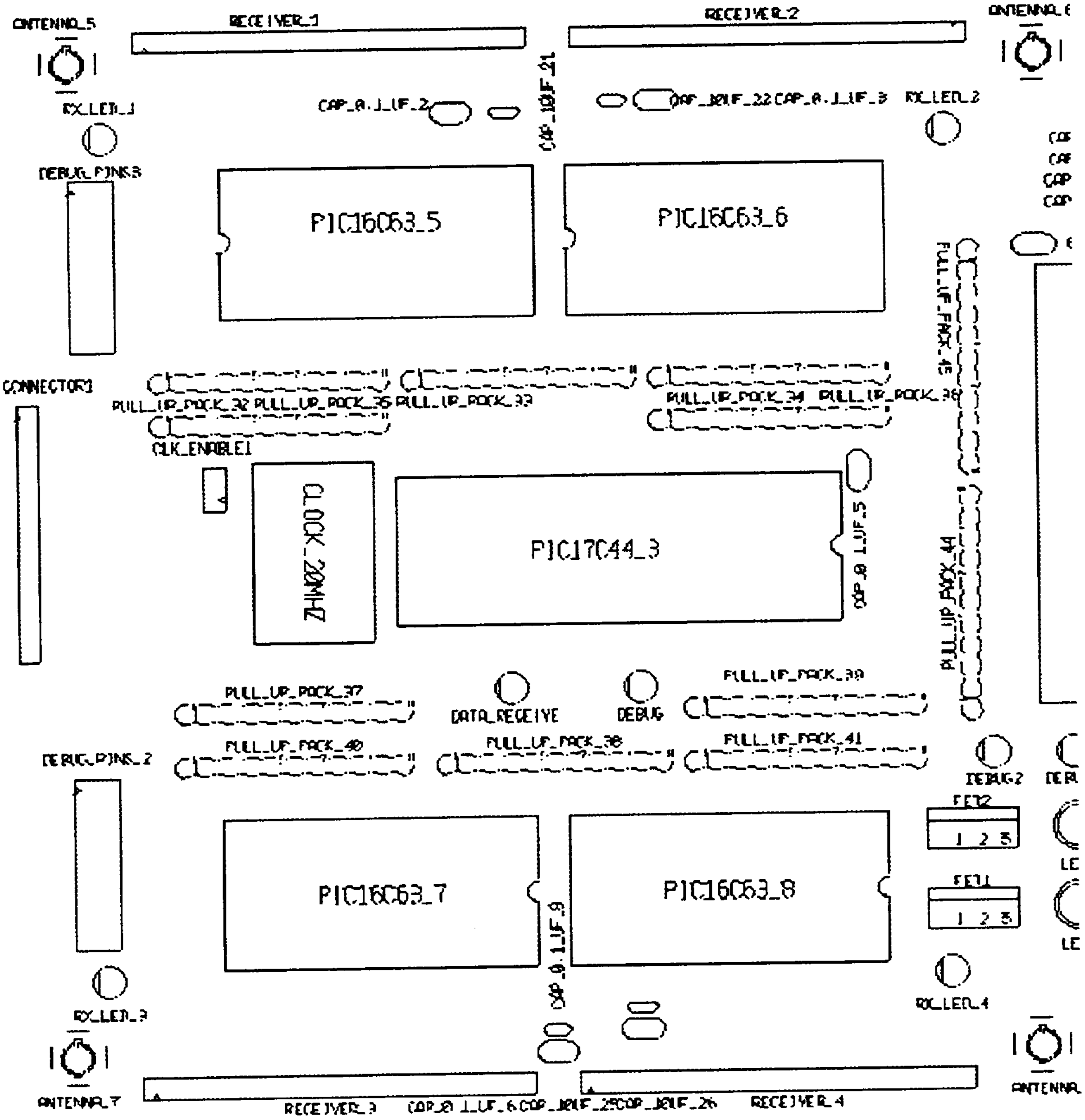


Figure 14. On-Board Output Controls

It is capable of stepping four stepper motors through the stepper motor output section (Fig. 15) and driving two motors through pulse-width-modulation. The purpose of the stepper motor controllers is for the on-board mass balancing that is to take place in the future. The PWM outputs are intended for the DC fan control with an extra PWM output for another DC motor control. The master output controller receives its output information from the command computer synchronously from the on-board master receiver controller.

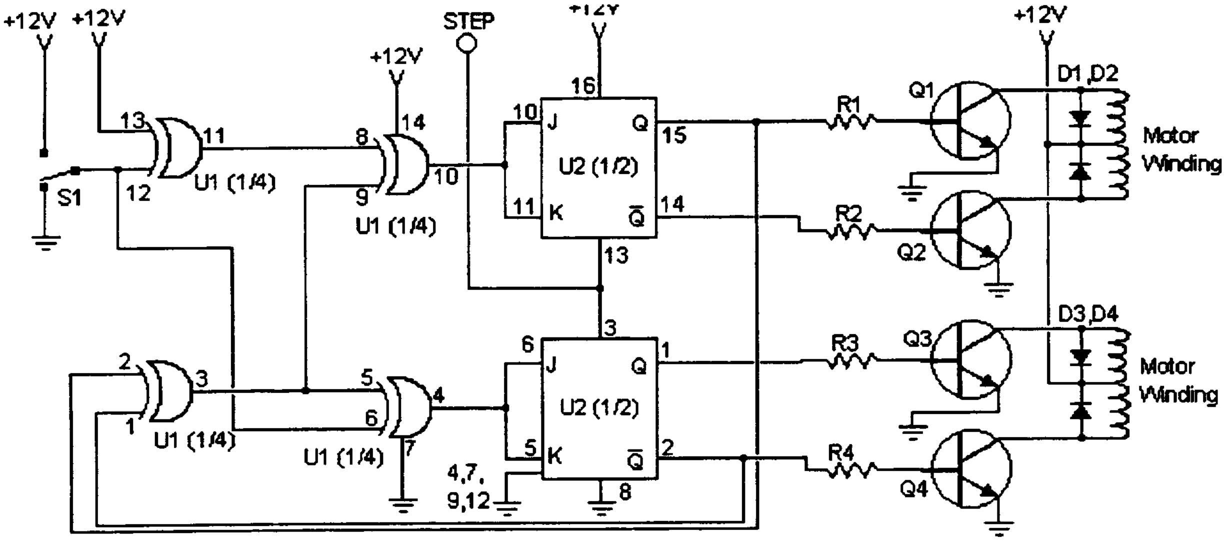


Figure 15. Stepper Motor Controller Circuit Schematic

The Command-Side Data Reception Subsystem

The command-side data reception system (Fig. 16) is comprised of five microcontrollers and four 900 MHz high-speed receivers.

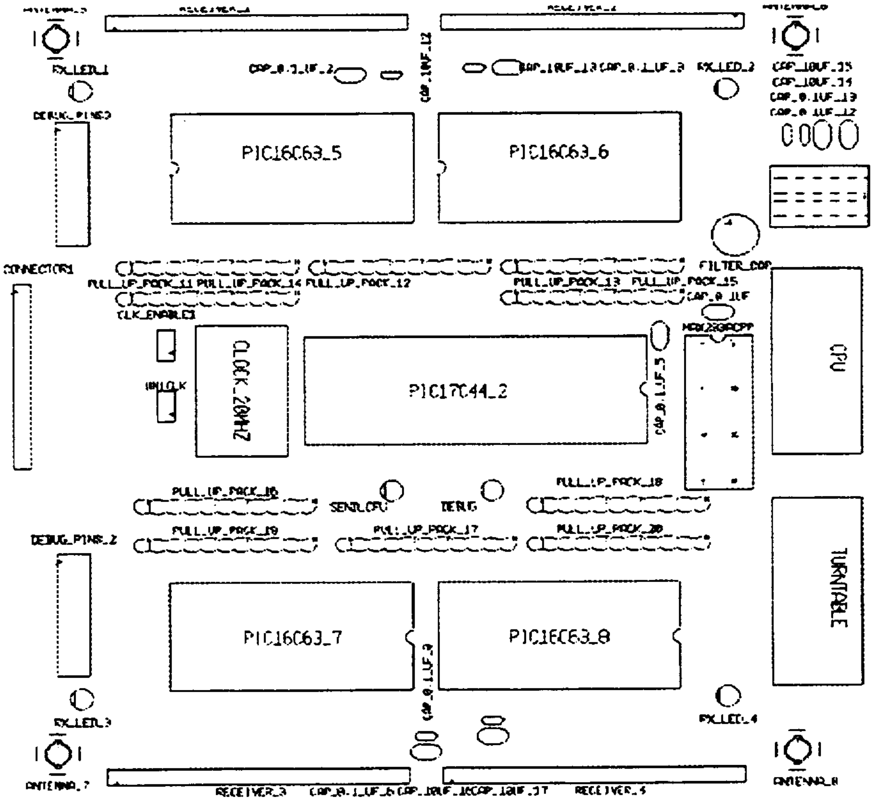


Figure 16. Command-Side Data Receiver

The purpose of the system is to receive the incoming spacecraft model data from the on-board transmission subsystem. It then asynchronously sends the information to the computer via the MAX232A RS-232 driver. The four receivers capture the transmitted data and asynchronously pass the data to the four receiving PIC16C63A microcontrollers. The receiving microcontrollers then cue the output master controller, and, under the direction of the receiving master microcontroller, pass the data through the connected bus to the master receiver microcontroller.

The Command-Side Data Transmission System

The command-side data transmission system (Fig. 17) is made up of five microcontrollers and four 900 MHz high-speed transmitters. Its purpose is to transmit the data collected from the devices on board to the command side receivers, which will deliver the data to the computer for analysis. Each of the four transmitters is modulated asynchronously by a PIC16C63A microcontroller. The transmitting microcontrollers are provided with transmission data from a PIC17C44 master microcontroller, which places the data a common bus and negotiates the data acknowledgement through two control lines to each of the four transmitting microcontrollers. The master microcontroller receives its information from the master controller in the data acquisition/input stage of the on-board subsystem.

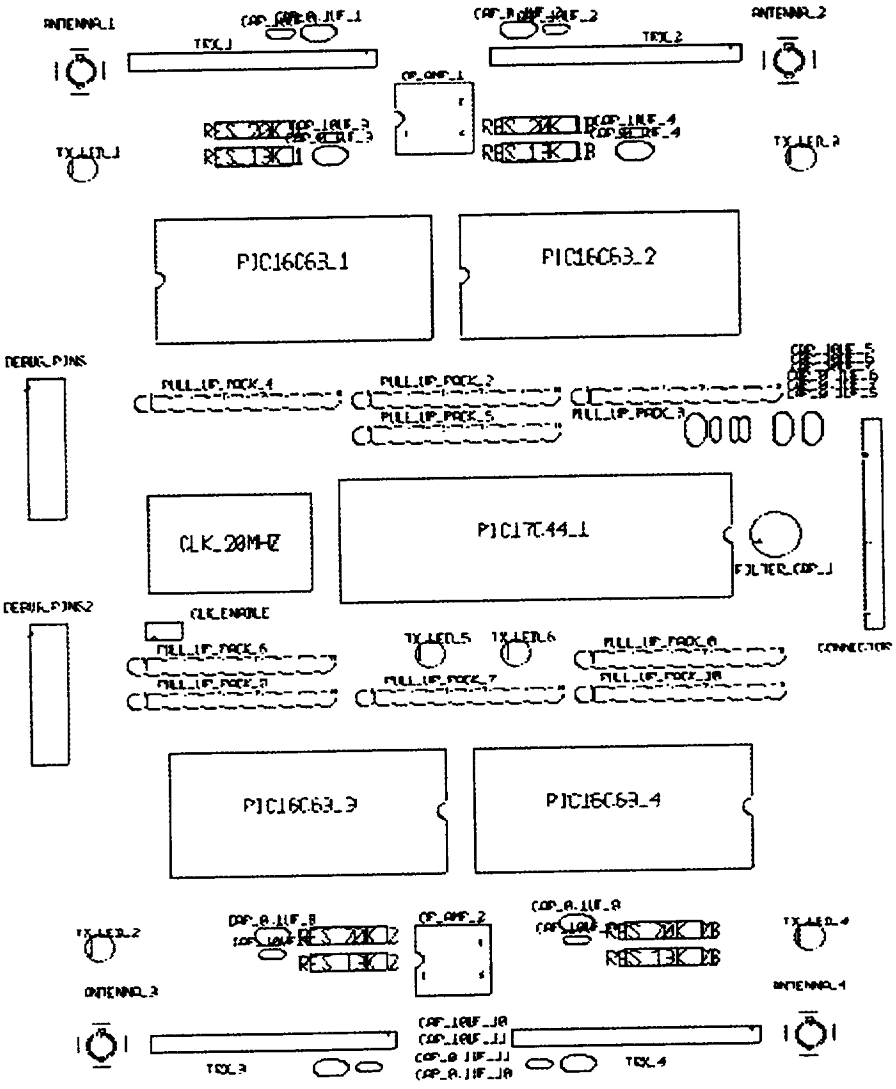


Figure 17. Command-Side Transmitters

Printed Circuit Board Layout and Fabrication

The layout process was extremely involved. It consisted of placing components, creating new components, assigning individual pins to their appropriate nets, spacing the components properly, routing and sizing ground and supply lines correctly, and manually routing traces which the autorouter could not route itself. This process took the longest portion of the project time. After completing the layout design, the board files were electronically sent to a company which then fabricated the boards. Following board fabrication, several hours were spent hand soldering the individual components to the pc-boards. During the component placement and initial testing, several layout errors were found and corrected.

CONCLUSIONS

An interdisciplinary team of undergraduate engineering and science students and faculty at Oral Roberts University is pursuing the development of an experimental apparatus to test the coning stability of spinning spacecraft under thrust. The integration of several engineering subsystems is required to successfully accomplish this objective. Mechanical systems for accelerating an air bearing that floats a spacecraft model are controlled based on model attitude data from an optical attitude sensing system. Communication with systems on-board the spacecraft model is accomplished via a two-way RF data link. Software remains to be developed for communications systems and the management of data taken during testing. Full integration of these systems is currently underway.

REFERENCES

1. G. A. Flandro, M. Leloudis, and R. Roach, "Flow Induced Nutation Instability in Spinning Solid Propellant Rockets, Final Report," AL-TR-89-084, 1990.

2. D. M. Halsmer and D. L. Mingori, "Nutational Stability and Passive Control of Spinning Rockets with Internal Mass Motion," *J. of Guidance*, Vol. 18, No. 5, Sept.-Oct., 1995, pp. 1197-1203.

3. Dominic M. Halsmer and Amy R. Fetter, "Optimal Design of a Passive Coning Attenuator for Spinning Spacecraft Under Thrust," AAS/AIAA Space Flight Mechanics Meeting, Monterey, Ca., Feb. 9-11, 1998.

4. R. X. Meyer, "Stability Tests of Spin-Stabilized Spacecraft in the Presence of Thrust," *J. of Guidance*, Vol. 14, No. 1, Jan.-Feb., 1991, pp. 206-208.

5. Dominic M. Halsmer, Amy R. Fetter, and Maduka C. Chidebelu-Eze, "Simulation Accuracy of an Apparatus to Test the Stability of Spinning Spacecraft Under Thrust," *Proceedings of the Flight Mechanics Symposium*, NASA Goddard Space Flight Center, Greenbelt, Md., May 1997, NASA CP-3345, pp. 1-15.

6. D. M. Halsmer, W. E. Bair, and P. Ng, "The Spinning Rocket Simulator: An Experimental Design Project for Teaching and Research," *ASEE Annual Conference Proceedings*, Washington, D.C., June 24-26, 1996.

REVIEW OF SPINNING SPACECRAFT DYNAMICS ANALYSES AND INFLIGHT EXPERIENCE[1]

Neil A. Ottenstein
Computer Sciences Corporation
Lanham-Seabrook, Maryland USA 20706

ABSTRACT

This paper presents investigations of attitude drift model results and bias trends for the Global Geospace Science (GGS) Interplanetary Physics Laboratory (WIND) and the Polar Plasma Laboratory (POLAR) as well as a study of Sun-only attitude determination for POLAR.

INTRODUCTION

The Global Geospace Science satellites, WIND and POLAR, were launched on November 1, 1994 and February 24, 1996, respectively. The Flight Dynamics Facility has accumulated extensive flight data pertinent to attitude dynamics analysis and sensor performance for these missions.

Both spacecraft are spin-stabilized about their principal axes. Historical data are shown concerning the biases estimated as part of the attitude determination process for the two spacecraft. Seasonal changes in the biases are shown as well as improvements in attitude determination accuracy over time due to long-term bias calibration of the WIND sensors. This paper also compares and discusses differences between computed attitudes and attitude drift model results for both missions. Finally a study of Sun-only attitude determination using POLAR data is presented.

WIND SENSOR CALIBRATION

The primary sensors for WIND calibration are two digital Sun Sensor Assemblies (SSA), each with two redundant heads, and a Star Scanner Assembly (SSCA). Each of the two SSAs use two parallel heads for measurement of the angle between the Sun and the spin axis. Each head provides a digital measurement of the Sun angle in a plane with a 128-degrees field of view (FOV). The two SSAs are mounted such that Sun angles are observable in the range 5 to 175 degrees, with an accuracy of better than 0.1 degrees in the range 70 to 110 degrees. These sensors also provide the Sun crossing time with an error of about 0.75 ms (at 20 rpm) to be used as a reference time for the spin period and Earth and star crossing times. So far the SSAs have performed better than their specification for both WIND and POLAR. The SSCA contains two V-slit sensors. It provides star crossing times for each slit accurate to 0.0035 ms at 20 rpm.

Task personnel calibrated the WIND attitude sensors with the Multimission Spin-Axis Stabilized Spacecraft (MSASS) Flight Dynamics Support System (FDSS) Differential Corrector (DC) subsystem. DC provides a number of biases for each sensor that can be determined and can solve for one, two, or no attitudes. During the normal phase of the WIND mission, analysts were interested in the SSAs mounting angle bias with respect to spin axis

[1] This work was supported by the National Aeronautics and Space Administration (NASA) / Goddard Space Flight Center (GSFC), Greenbelt, Maryland, under Contract GS-35F-4381G, Task Order no. S-24280-G.

NASA/GSFC, Guidance, Navigation and Control Center, *Flight Mechanics Symposium*, Greenbelt, MD USA, May 1999.

(Sun angle bias), the SSCA's mounting angle bias with respect to spin axis, the SSCA azimuth bias with respect to Sun crossing, and the SSCA tilt with respect to spin axis.

The spin axis attitude determination requirements are 1.0 degrees (3σ) for real-time attitude determination during the early mission and maneuver support for WIND. For the normal mission 0.5 degrees (3σ) accuracy is required for WIND.

During the normal mission phase (post deployments), WIND personnel updated the calibration as needed or whenever the attitude changes. Due to observability, they were not been able to solve for all the biases and have focused primarily on the Sun angle biases. After initial calibration the nominal solutions incorporated a constant star sensor mounting angle bias of 0.005 degrees and a Sun-to-star azimuth bias of negative 0.1176 degrees in the sensor calibration file for each run. Figure 1 shows the Sun angle biases for each WIND SSA from January 5, 1995 through June 19, 1997. Examine the first part of this figure through July 1995. During the early part of the normal mission phase, the biases started out small with not much change, but over this time they developed a pronounced drift. This drift was believed to be due to biases with the star sensor that were not incorporated in the nominal bias calculations which only solved for attitude and Sun angle biases. Using the daily account of measured Sun angles for each sensor, star angle and Sun-star dihedral angle up through June 17, 1995 a better calibration of the biases could be produced. This was done using each Sun sensor observation and dot product calculations of the observables.

The results were as follows:

SSA1 and SSCA:

Sun angle bias	= -0.160 deg
scanner mounting angle bias	= 0.149 deg
scanner azimuth bias	= -0.149 deg

RMS of Sun-to-star angle residuals

before calibration:	0.204 deg
after calibration:	0.007 deg

SSA2 and SSCA:

Sun angle bias	= -0.019 deg
scanner mounting angle bias	= 0.157 deg
scanner azimuth bias	= -0.177 deg

RMS of Sun-to-star angle residuals

before calibration:	0.089 deg
after calibration:	0.005 deg

There was no estimate for the uncertainties of these biases, but the uncertainties are in the ratio of 1.00:0.70:5.00 for the Sun angle, star scanner mounting angle and star scanner azimuth biases respectively. Thus, the azimuth bias has the largest uncertainty by a factor of 5 or more.

Since the first Sun sensor is the primary one, the bias values of the Star mounting angle and azimuth in the first set of results were taken into the sensor calibration file for future calculations. The results of these calculations were quite dramatic as seen in in the latter part of Figure 1. Figure 2 shows just these biases since the calibration was introduced in the solutions on June 27, 1995 through October 13, 1997 which is about as far as the biases have

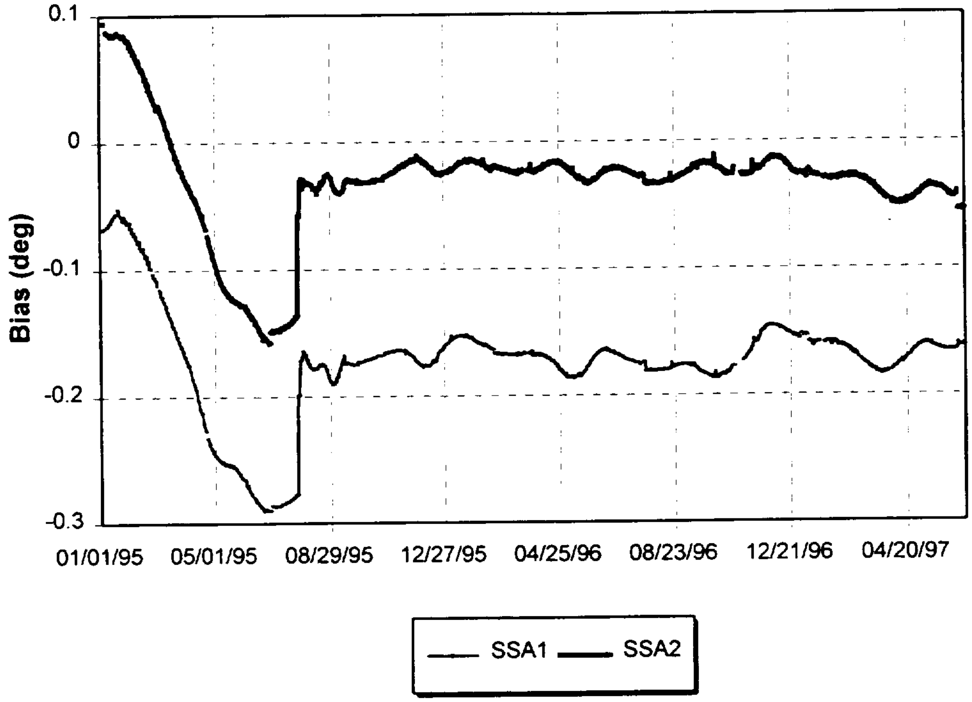


Figure 1. WIND SSA Bias History

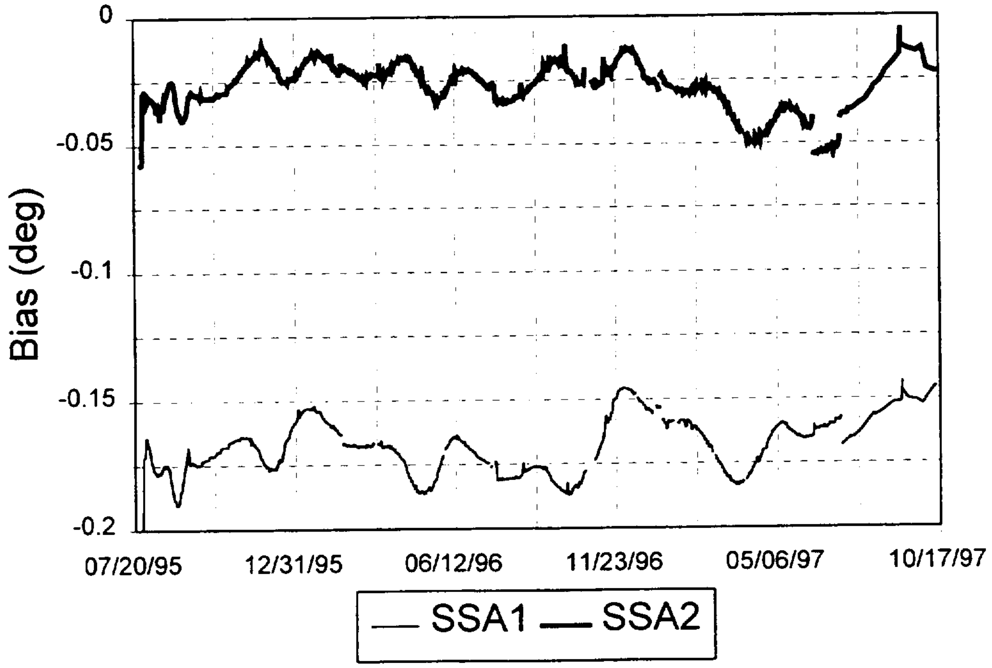


Figure 2. WIND SSA Biases After Calibration

currently been documented. While previously the Sun angle biases changed over 0.2 degrees over six months, since incorporation of the new SSCA biases the spread of Sun angle biases was only approximately 0.05 degrees over more than two years. As seen, there is probably some seasonal bias still in the data, but this is much smaller than previously seen. Attitude solutions before and after the new biases differed by roughly 0.11 degrees, still well within the accuracy requirements. Any future changes should modify the attitude by even less.

WIND ATTITUDE DRIFT

There were several pre-mission drift analyses performed that were to characterize the expected attitude drift of the WIND satellite mainly due to solar pressure torques (References 1-2). The goal of these was to see whether additional attitude maneuvers would be required for WIND outside of any corrections due to changes caused by orbit maneuvers. The WIND satellite spin axis was required to be pointed within one degree of the South Ecliptic pole with one degree of control. In addition there was an attitude request on behalf of the Transient Gamma Ray Spectrometer (TGRS) instrument. This was a thermal requirement such that the spin axis must be controlled to maintain the Sun angle between 89.65 to 91 degrees. These analyses examined various starting attitudes and concluded that a maneuver would be required within four months at best. Furthermore analysis showed that on average the satellite would drift between 0.005 and 0.0065 degrees a day.

Figure 3 shows the observed daily angular drift using a calculation of one day's attitude solution from the day's before. On average this drift rate has been 0.007 degrees per day.

Figure 4 shows the attitude history of WIND through October 20, 1997 with right ascension plotted against declination. The attitude has been clearly drifting in a semi-circular pattern away from the South Ecliptic Pole. Depending on which quadrant with respect to the Pole the attitude starts, the pattern changes. The various non-smooth changes in attitude are due to the various maneuvers over the life of the mission.

Figure 5 shows the Sun angle history and how well the Sun angle requirement has been able to be obeyed. This Sun angle is the average of the SSA1 and SSA2 Sun angle readings. These are the raw values without incorporating the biases. Adding the biases in would lower the values by a bit less than 0.1 degrees. The history of the Sun angle readings is reproduced quite well time after time.

Figures 6, 7 and 8 show the attitude and Sun angle predictions covering nearly a four-month period between attitude maneuvers in early 1995. These predictions are from the solar pressure torque model. The actual solutions follow the general direction of the prediction. The right ascension begins to diverge after about a month and a half, while the declination and Sun angle break away earlier. It is to be noted that there were orbit maneuvers during this time period. Still the model is just too simple to accurately predict what happens. The best predictor that could be used for the future would be past results.

POLAR ATTITUDE AND BIAS RESULTS

Similar to WIND, the onboard sensors for POLAR include two horizon sensor assemblies (HSAs) and two SSAs. Unlike WIND, the SSAs are co-located on the spacecraft body. The nominal attitude for POLAR is with the +Z axis pointing within 1 degree of either positive or negative orbit normal, while keeping the sun angle greater than 87 degrees. The sun angle requirement necessitates a 180 degrees attitude flip every six months for the duration of the mission lifetime. The HSAs have full orbit coverage at the nominal attitude.

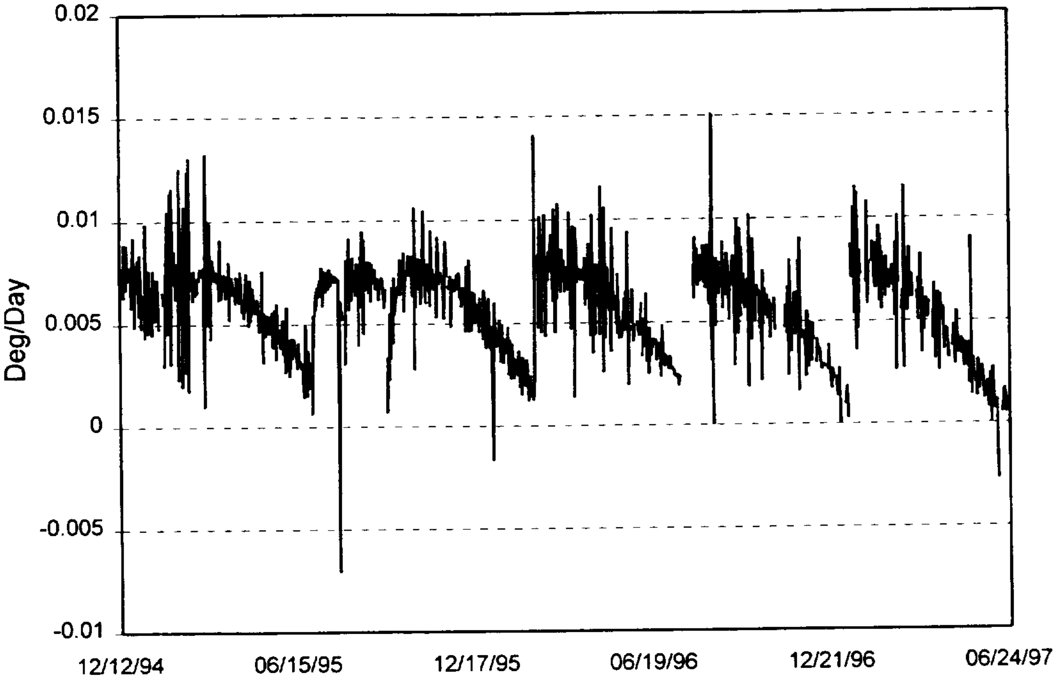


Figure 3. WIND Daily Angular Drift

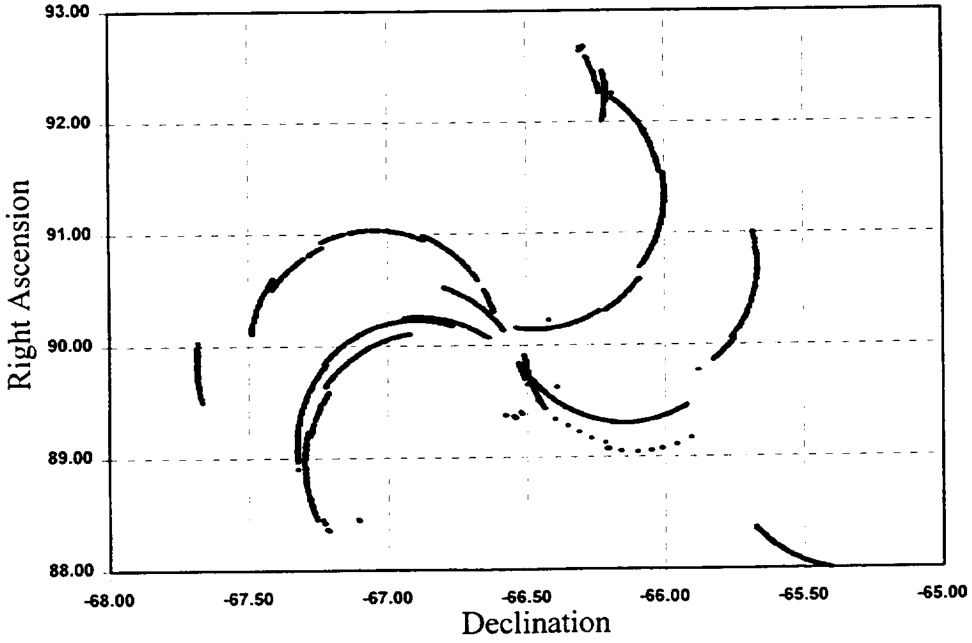


Figure 4. WIND Attitude History Through 10/20/97

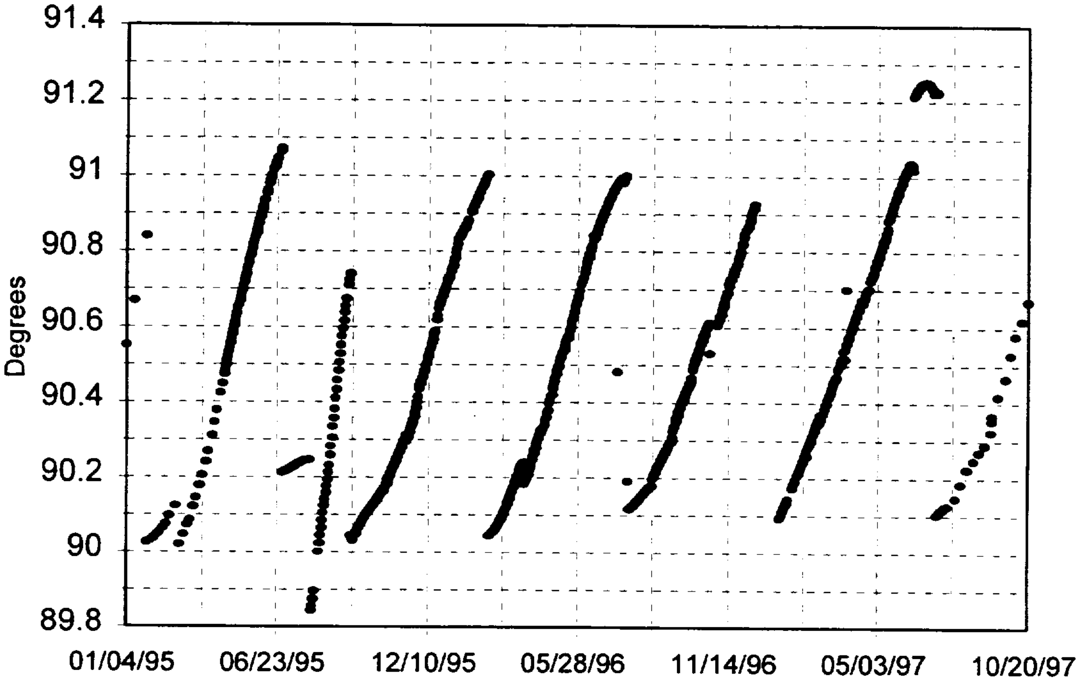


Figure 5. WIND Sun Angle History

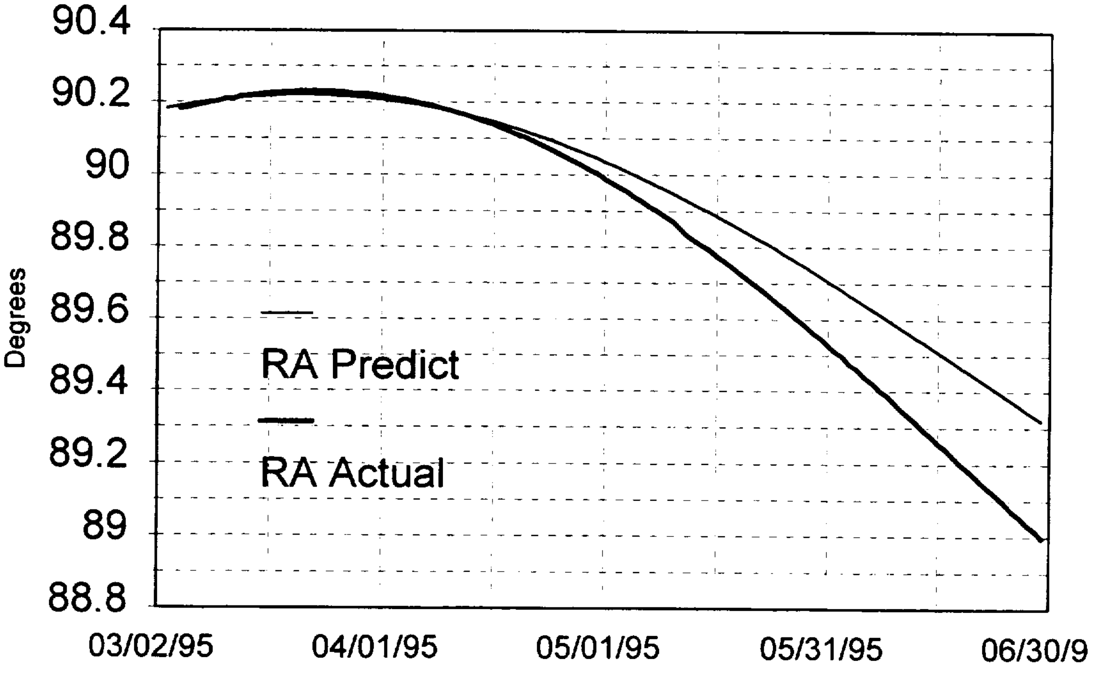


Figure 6. WIND Right Ascension Prediction and Actual

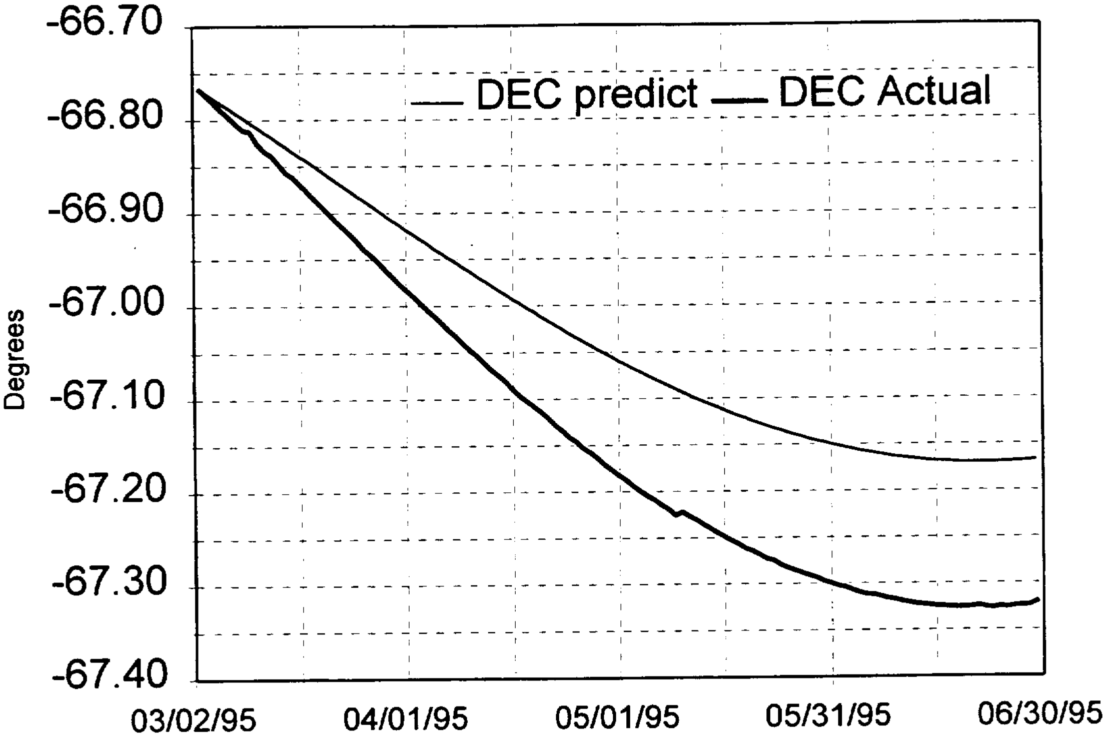


Figure 7. WIND Declination Prediction and Actual

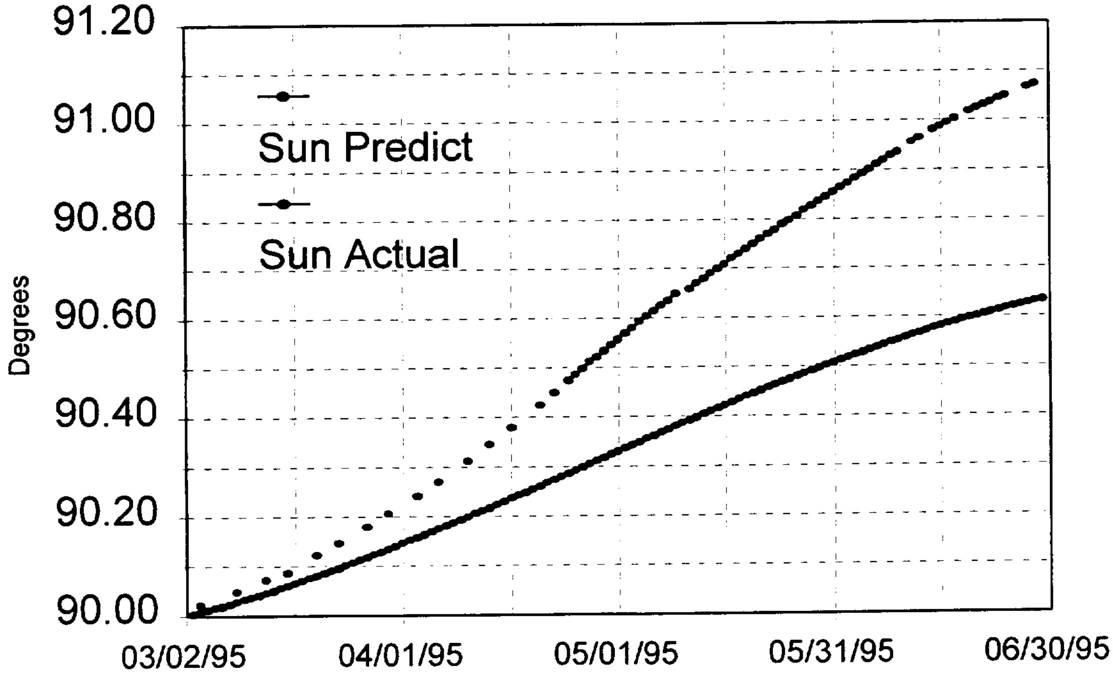


Figure 8. WIND Sun Angles Prediction and Actual

The spin axis attitude determination requirements for POLAR are also 1.0 degree (3σ) for real-time attitude determination during the early mission and maneuver support. For the normal mission 0.2 degrees (3σ) accuracy are required.

Figure 9 shows the SSA biases from March 18, 1996 through September 9, 1997. Some of the scattering and jumps in bias values coincide with the despun platform activity, which would affect the body coning angle and thus the computed biases. Maximum and minumum values appear to follow a seasonal pattern. The first maximum is around July 19, 1996. The minimum following that is around January 16, 1997 and the next maximum is around July 17, 1997. These dates are close to the halfway point between attitude maneuvers. The attitude was pointing near negative orbit normal from April 19, 1996 through October 17, 1996 with the halfway point being July 17. The attitude was pointing near positive orbit normal from October 19, 1996 through April 11, 1997 with the halfway point being January 14. The attitude was again near negative orbit normal from April 14, 1997 through October 10, 1997 with the halfway point being July 13.

Figure 10 shows the HSA biases from March 18, 1996 through September 9, 1997. The HSAs on POLAR experience a time delay due to the onboard electronics check for Moon interference with the Earth sensor. This uncompensated delay causes most of the observed azimuth biases. Again there is a clear seasonal dependence. The minimum biases in this figure appear to occur slightly after the peaks of the SSA biases. They are around July 29, 1996, January 28, 1997 and then July 25, 1997. Again there is a strong correlation with the halfway point of the attitudes.

Figure 11 shows the offsets of the attitude from orbit normal. The target attitudes were chosen to compensate for the change of the orbit plane between maneuvers. Thus, after a maneuver the difference is a local maximum value which decreases linearly over the next month or so until it attains a minimum value and then increases again. The minimum values in this time frame are August 8, 1996 and January 9, 1997 which are close to the halfway points between maneuvers and reinforces some of the correlation of biases.

Figure 12 shows a typical progression of right ascension during a positive orbit normal sequence. The attitude slowly drifts about 0.3 degrees from 113.7 to 114 degrees over the course of six months. Figure 13 shows the case for a negative orbit normal example. There appears to more more of a drift decreasing 0.45 degrees from 295.9 to 295.45 degrees over six months.

Figure 14 shows a typical behavior of positive orbit normal declination over six months. It starts at about –4.2 degrees, increases to –3.7 degrees and then decreases again to –4.2 degrees. Figure 15 shows a similar pattern for a negative orbit normal declination. It starts at about 3.4 degrees, increases to 3.9 degrees and then decreases back to about 3.5 degrees.

Figures 16 and 17 show a comparison over two months between attitude predictions (References 1, 3-5) and near real time solutions for the right ascension and declination. The change in attitude follows the drift predictions fairly well. Due to this success, instead of supplying a constant attitude prediction product, the drift predictions were used for the attitude prediction product. Furthermore, operations staffing time was reduced to take the near real solutions at less frequent intervals.

POLAR SUN ONLY STUDY

A study was performed on POLAR data using the MSASS attitude determination system to get a feeling for how accurate Sun-only attitude determination could be using various data spans. Data was accumulated for a series of 6 POLAR passes of roughly a half-hour in length with each series spanning nearly a whole day. Attitude determination was performed with a set of a priori attitudes to find out how close the attitude solution could come to the solution that was found in daily operations using Sun and Earth observations. This truth attitude based on the

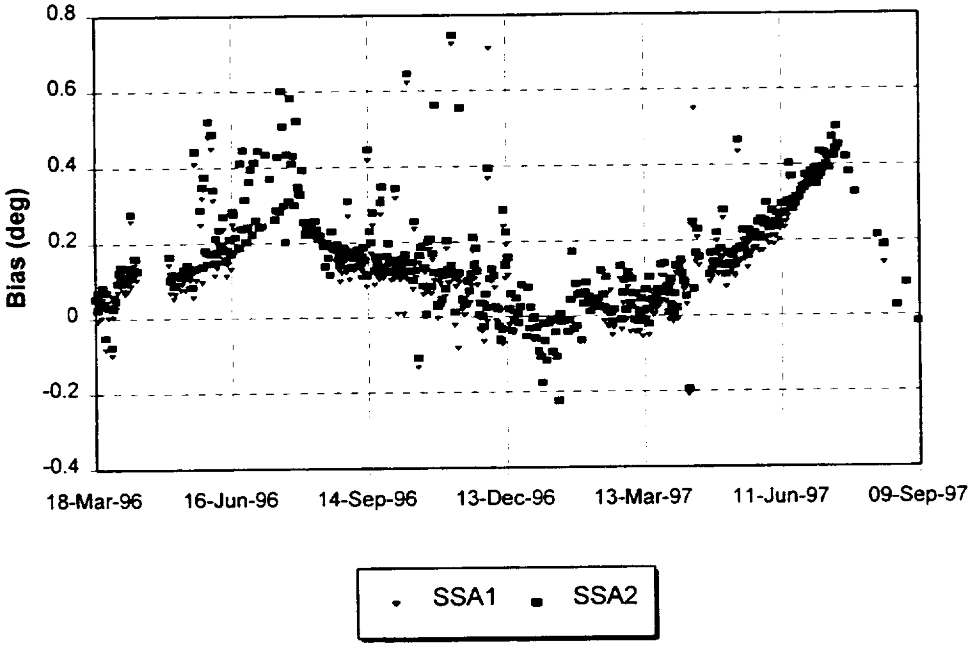


Figure 9. POLAR SSA Biases

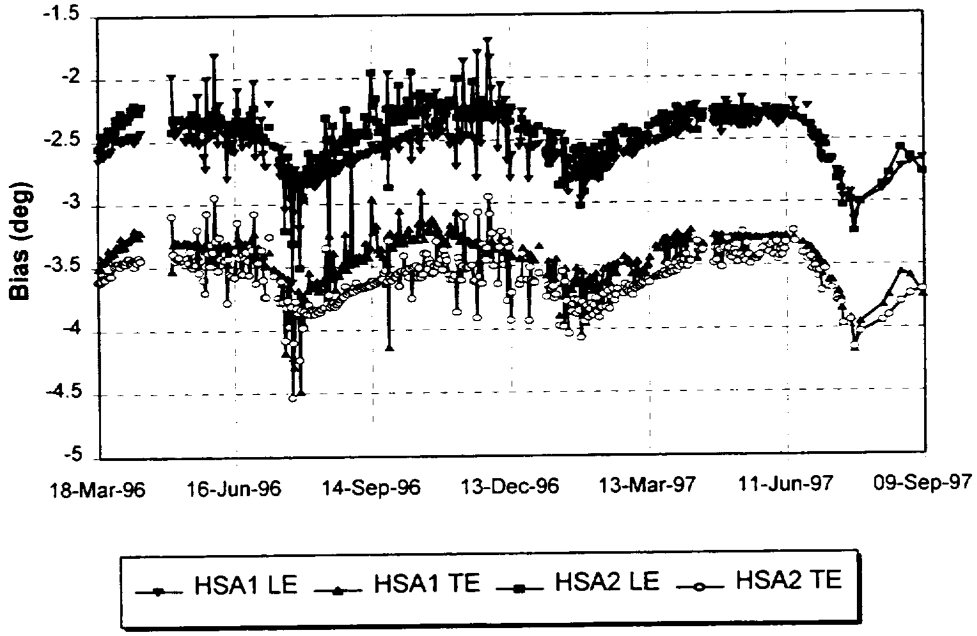


Figure 10. POLAR HSA Leading and Trailing Edge Biases

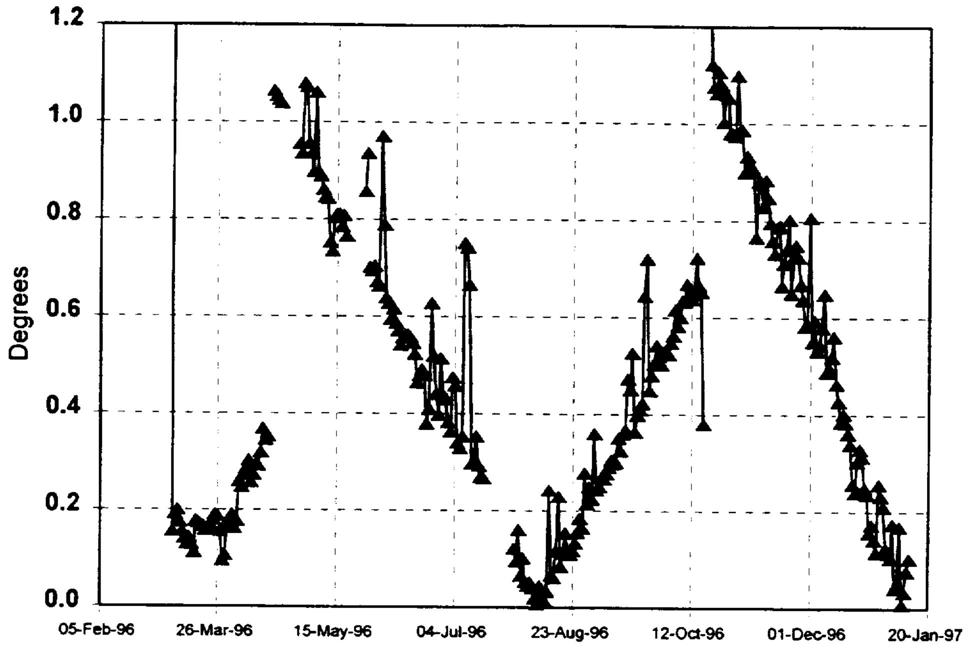


Figure 11. POLAR Attitude Offset from Orbit Normal

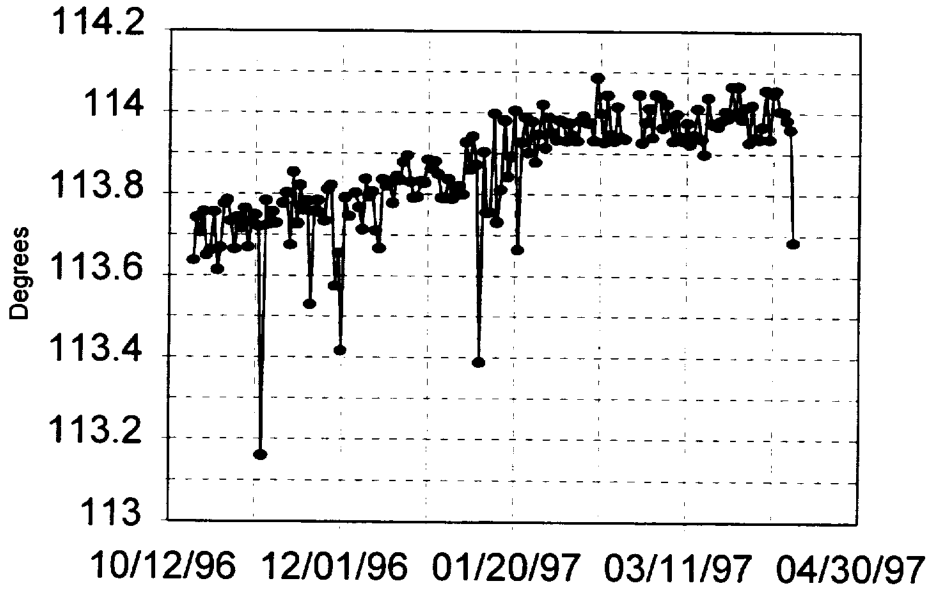


Figure 12. POLAR Positive Orbit Normal Right Ascension

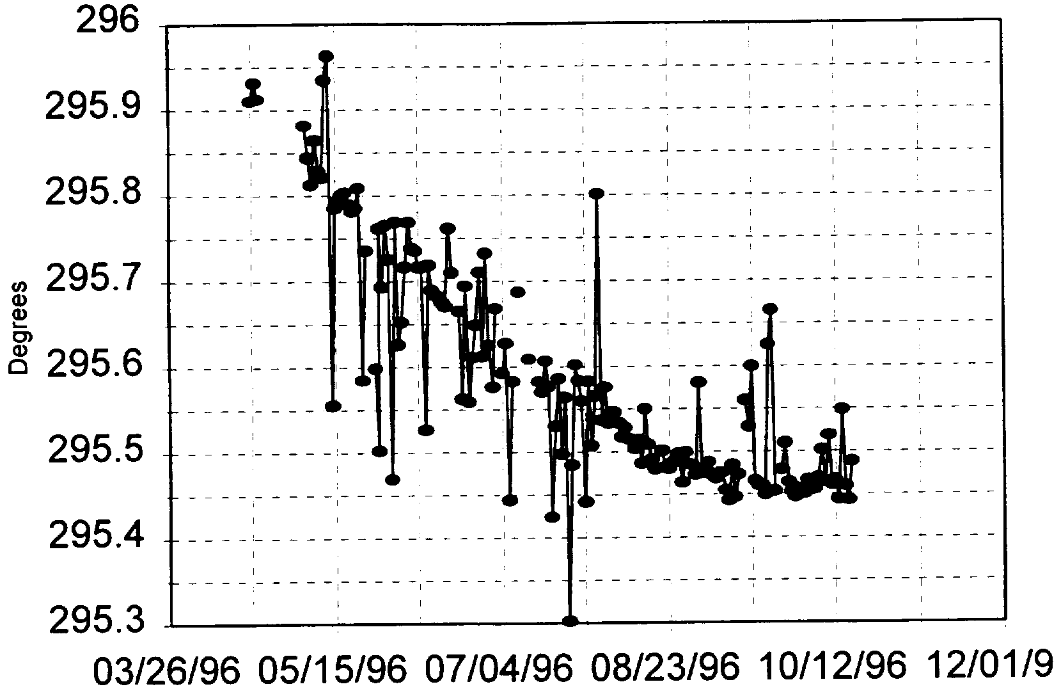


Figure 13. POLAR Negative Orbit Normal Right Ascension

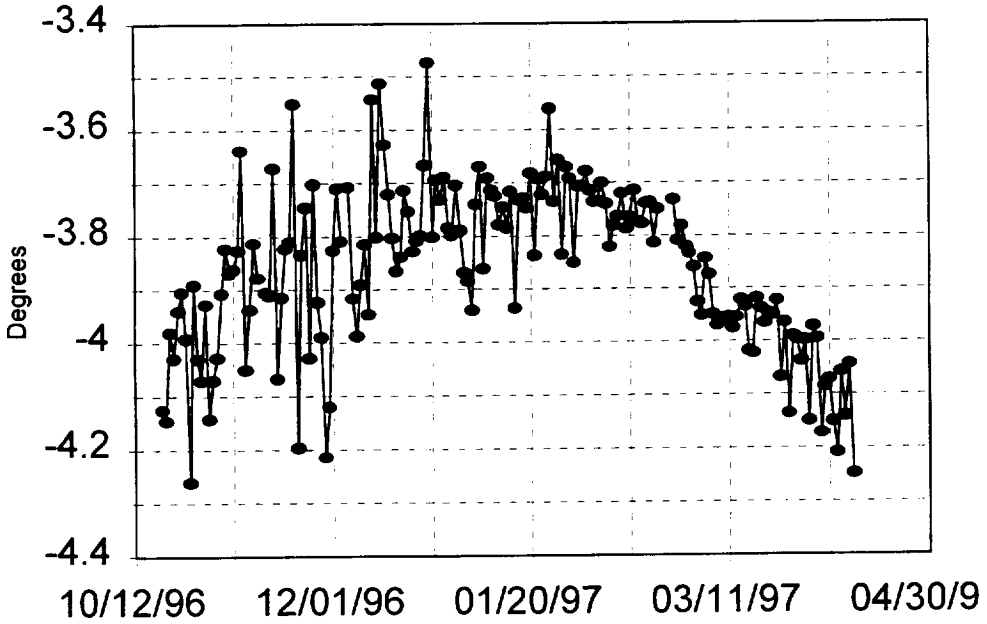


Figure 14. POLAR Positive Orbit Normal Declination

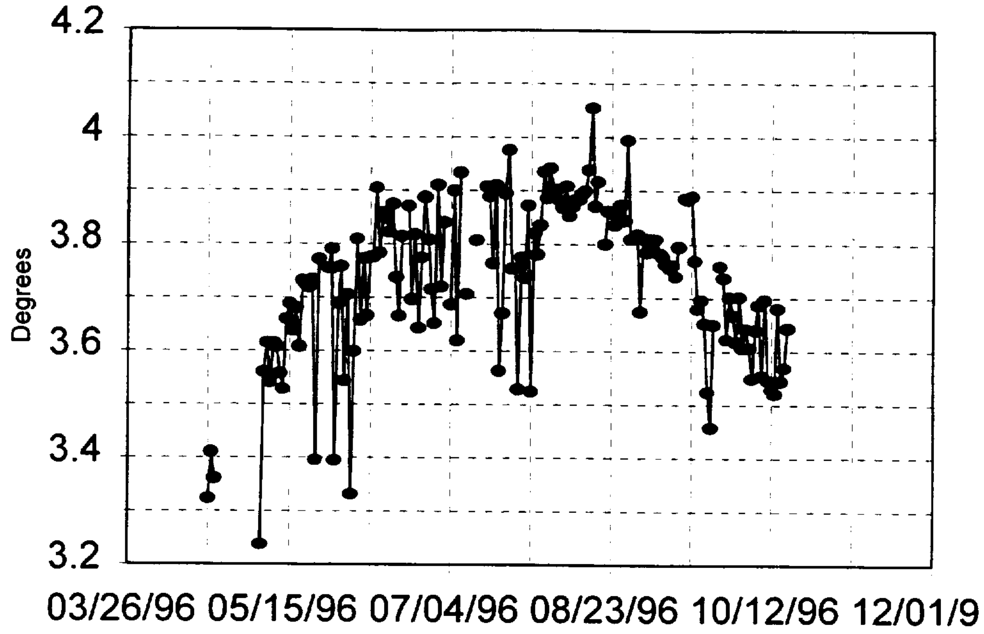
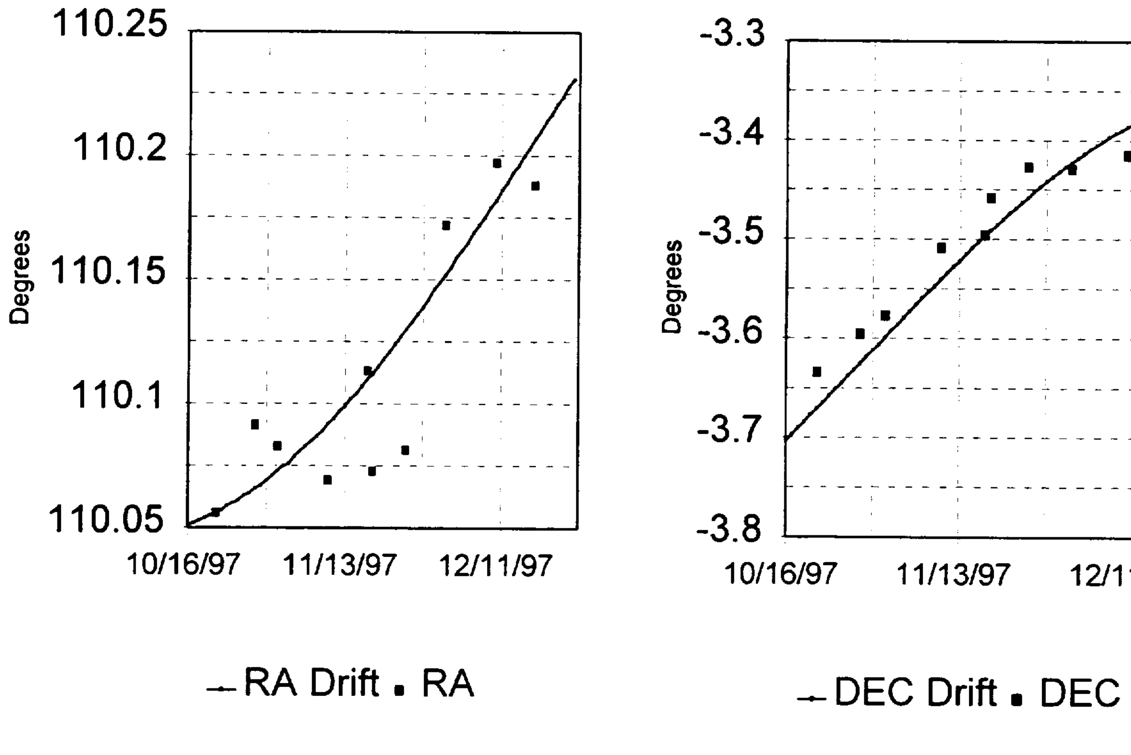


Figure 15. POLAR Negative Orbit Normal Declination



Figures 16 & 17. POLAR Right Ascension and Declination Drift Prediction and Near Real Time Results daily solutions using Sun and Earth observations was right ascension (RA): 287.7 degrees, declination (DEC): 4.08 degrees. This truth attitude itself is expected to be within the required 0.2 degrees (3σ) accuracy.

The data used in the study:

Pass Number	Start Time	Duration	Approx. Sun angle (deg)
1	7/17/98 03:42	34 min	153.02
2	7/17/98 07:08	34 min	152.99
3	7/17/98 10:19	25 min	152.96
4	7/17/98 13:29	34 min	152.94
5	7/17/98 21:52	34 min	152.89
6	7/18/98 01:07	34 min	152.86

The Sun angle values fluctuate slightly above the angle given above.

The a priori attitudes are the set of test cases. Both attitude and Sun angle bias were determined in all solutions. The distance is the value obtained by taking the arc cosine of the dot product of the "truth" attitude and the a priori attitude or the solution.

The results are given starting with six passes and then proceeding down to just one pass. As seen when a solution could be obtained, the results for all using three passes or more were all within a half degree of the "truth" solution even for a starting a priori attitudes of more than 5 degrees away from the expected "truth" attitude. This may be sufficient accuracy for missions with less stringent attitude knowledge requirements than POLAR, reducing the need for other sensors. There were some problems with using three passes that did not occur when using four passes, but overall it appears that just three passes of data spread throughout 7 hours seems to be adequate to obtain a Sun-only solution in most cases.

Using only one pass, there was not enough observability to solve for both the right ascension and declination successfully. The declination usually approached the "true" declination more successfully than the right ascension did. There were also more problems in obtaining a solution as the data was rejected more frequently. When using two passes, matters were even worse, though, probably because the Sun angle just didn't change enough over the limited time between the two passes. Solutions either did not converge or went off to nonsensical solutions.

Following are some tips and tricks that were necessary for obtaining the Sun-only solutions:

1) The Sun model rejection tolerance was increased from 2 degrees to 2.5 degrees in order to accept the data for distant a priori attitudes.

2) If a solution could not be initially obtained, the data could be rerun without updating the state vector. Since the Sun angle appears to be more sensitive to changes in declination, the Sun model residual average could then be subtracted from the a priori declination and a new a priori attitude could be used for another try.

3) The assumed error in a priori right ascension and declination was increased after an unsuccessful run from 2 to 4 degrees to produce a successful run.

One final note in using MSASS, when either the right ascension or declination is close to 0 degrees the program is supposed to go into a rotated coordinate system so that solutions can continue without a division by 0. Unfortunately, in this implementation of MSASS, an error occurs when rotating back from the converged solution. If this could be corrected there would have been more success in various runs.

The attitude solutions for the test cases follow:

6 passes: goes through 22 hours after first data point

a priori RA	a priori DEC	a priori distance	RA solution	DEC solution	Solution distance
287.778	4.2	0.143	287.376	4.32	0.403
287.778	2.2	1.88	287.404	4.24	0.336
285.778	2.2	2.686	287.401	4.25	0.343
285.778	6.2	2.856	287.368	4.34	0.421
289.778*	6.2	2.963	287.373	4.33	0.411
291.778	6.2	4.581	287.379	4.31	0.394
291.778	7.2	5.119	287.376	4.32	0.403
291.778	8.2	5.780	287.373	4.23	0.359
285.778	8.2	4.541	287.366	4.35	0.429

4 passes: goes through 10 hours and 20 min after first data point

a priori RA	a priori DEC	a priori distance	RA solution	DEC solution	Solution distance
287.778	4.2	0.143	287.366	4.258	0.377
287.778	2.2	1.88	287.455	3.998	0.343
285.778	2.2	2.686	287.4436	4.022	0.262
285.778	6.2	2.856	287.334	4.348	0.453
289.778*	6.2	2.963	287.376	4.233	0.358
291.778	5.02	4.172	287.417	4.12	0.285
291.778	6.2	4.581	287.391	4.195	0.329
291.778	7.2	5.119	287.379	4.233	0.355
291.778	8.2	5.780	287.370	4.256	0.373
285.778	8.2	4.541	287.326	4.375	0.475

*The run failed when the declination was sent to 0. It was run again with the state vector not updated. This produced residuals of 1.18 degrees. This value was subtracted from the declination and then the final solution was found.

3 passes: goes through 7 hours and 2 minutes after first data point

a priori RA	a priori DEC	a priori distance	RA solution	DEC solution	Solution distance
287.778	4.2	0.143	287.558	4.347	0.302
287.778*	2.2	1.88	287.556	4.35	0.306
285.778	2.2	2.686	287.63	4.08	0.070
285.778	6.2	2.856	287.52	4.44	0.402
289.778**	6.2	2.963	287.56	4.33	0.286
291.778	6.2	4.581	287.58	4.29	0.242
291.778	7.2	5.119	287.57	4.33	0.282
291.778	8.2	5.780	Unable to get solution		

*All observations were rejected, but with –2.1 residuals. So subtracted residual from declination to run again with final result.

**The run actually failed when the declination was sent to 0. It was run again with the state vector not updated. This produced residuals of 1.17 degrees. This value was subtracted from the declination and then the final solution was found.

2 passes: goes through 4 hours after first data point

a priori RA	a priori DEC	a priori distance	RA solution	DEC solution	Solution distance
287.778	4.2	0.143	none		
287.778	2.2	1.88	287.27	4.095	0.429

All other attempts with 2 passes either did not converge or went to a nonsensical solution.

1 passes

a priori RA	a priori DEC	a priori distance	RA solution	DEC solution	Solution distance
287.778	4.2	0.143	287.488	4.315	0.316
287.778	2.2	1.88	287.116	3.90	0.610
285.778	2.2	2.686	285.9975	3.61	1.764
285.778*	6.2	2.856	286.593	4.02	1.106
289.778**	6.2	2.963	288.393**	4.43	0.773
291.778	5.02	4.172	289.264	4.79	1.712
291.778***	6.2	4.581	290.01***	4.99	2.477
291.778***	7.2	5.119	289.97***	5.01	2.446

* The run actually failed when the declination was sent to 0. It was run again with the state vector not updated. This produced residuals of 2.33 degrees. This value was subtracted from the declination and then the final solution was found.

**All observations were rejected in the sixth iteration. It was run again with the state vector not updated. This produced residuals of 1.18 degrees. This value was subtracted from the declination and another attempt was done, but it did not converge. One of the solutions it was switching between is given here.

***All observations were rejected after iteration #5, which is given as the solution here

In conclusion, missions with less stringent spin axis attitude knowledge requirements than POLAR may be able to use a Sun only attitude determination method, thus reducing the need for other sensors.

REFERENCES

1. Goddard Space Flight Center, Flight Dynamics Division, 554-FDD-91/154, *ISTP/GGS WIND and POLAR Missions Attitude Drift Rate Analysis*, D. Niklewski (CSC) and N. Ottenstein (CSC), November 1991.

2. Computer Sciences Corporation, CSC 53708 17, “WIND Spin Axis Attitude Drift,” (Memorandum), N. Ottenstein, July 14, 1993.

3. Computer Sciences Corporation, CSC 53708 15, “POLAR Attitude Drift Study,” (Memorandum), R. Edmonds, July 14, 1993.

4. Computer Sciences Corporation, CSC 55708 09, “POLAR Attitude Drift Study,” (Memorandum), D. Niklewski, August 29, 1995.

5. Computer Sciences Corporation, CSC 56708 07, “POLAR Attitude Drift Study,” (Memorandum), D. Niklewski, January, 19, 1996.

SOHO RECOVERY

Jean-Philippe Olive
Matra Marconi Space at NASA-GSFC
e-mail: jolive@pop400.gsfc.nasa.gov

ABSTRACT

The Solar and Heliospheric Observatory (SOHO) is a joint program between the European Space Agency (ESA) and NASA. After a launch (by an Atlas IIAS rocket) on December 2 1995, SOHO was certified fully operational, at its vantage halo orbit around the first Lagrangian point, in April 1996. The instruments on-board SOHO brought a wealth of discoveries such as flows of gas inside the Sun; rivers of plasma beneath the surface of the sun; more than 50 sungrazing comets; spectacular images and movies of Coronal Mass Ejections which could allow to forecast space weather.

At the beginning of a 1-week long series of maneuvers, the control of the spacecraft was lost on June 25, 1998. Based on the last telemetry data received from SOHO, engineers had reasons to believe that the spacecraft was slowly spinning in such a way that its solar arrays do not receive adequate sunlight to generate power. However it appeared that SOHO's solar panels will be exposed to an increasing amount of sunlight each day as it orbits the Sun.

On July 23, the huge antenna of Arecibo confirmed SOHO spacecraft was at its predicted location and moreover than the spin rate was slower than 1 revolution per minute. On August 3, the spacecraft carrier was detected. As soon as one battery was charged up, a short burst of telemetry was collected on August 8.

The following challenge was to thaw the frozen hydrazine, starting with the tank, without discharging too much the batteries. When the tank was thawed after 11 days of heating, it appeared impossible to warm up the whole propulsion subsystem with heaters "digging into" the batteries during each eclipse period. Fortunately it was possible to patch the on-board software to use a solar array current like a “fake” thermistor, in order to switch ON heaters only when power was available from the solar arrays. This “sunheat” mode was used to increase the heating power without draining the batteries and had to be tuned several times to ensure the consumed power would stay within the available amount provided by the solar arrays. Such a tuning allowed us to put more and more heaters into “sunheat” mode and finally to thaw most of the propulsion subsystem.

On September 16, the spin rate was reduced by firing one of the thawed thruster and then a partial Emergency Sun Reacquisition successfully re-pointed the spacecraft, hence giving full power capability. Following a re-commission of the equipment and Station Keeping maneuver, SOHO was back into Normal Mode on Sept 25. Finally the instruments were tested and remarkably all work again correctly.

INTRODUCTION

The series of events that lead to loss of telemetry are presented in the SOHO Mission Interruption Final report (see Reference 1).

This paper is aimed at describing the recovery of the SOHO spacecraft, it covers mainly a 3-month period from June 25 to September 25 1998 when the attitude control was back to normal mode.

How telecommunications were reestablished with SOHO spacecraft is described in this paper, highlighting the involvement of engineers in Europe and at NASA GSFC, and also the support of the scientific community.

The long and careful thawing of the propulsion subsystem was a major challenge of the recovery and is sum up here after.

Using a solar array current like a "fake" thermistor is presented, including the different settings used before achieving a final "sunheat" power profile.

Finally the sun reacquisition maneuver is explained.

During normal operations SOHO always looks at the sun (X-axis), the two solar arrays give the Y-axis and the Z-axis is aligned with the poles of the sun. See SOHO spacecraft and axis given in figure 1. The two solar arrays always face the sun and can provide up to 1500 W. Power storage is ensured by two 20Ah NiCd batteries. In case of loss of sun pointing, an Emergency Sun Reacquisition (ESR) is triggered and then the Failure Detection Electronics (FDE) takes over with a hardwired attitude control using propulsion branch B thruster. The propulsion uses hydrazine as a mono-propellant with helium for pressurant. The eight 4N thrusters are used only during maneuvers (propulsion branch A) and in case of ESR (propulsion branch B).

SYMBOLS AND ABBREVIATIONS

ACS	Attitude Control Subsystem
AGC	Antenna Gain Control
AOCS	Attitude and Orbit Control Subsystem
BCR	Battery Charge Regulator
BDR	Battery Discharge Regulator
COBS	Central On-Board Software
DSN	Deep Space Network
EOC	End Of Charge
ESA	European Space Agency
ESR	Emergency Sun Reacquisition
FDE	Failure Detection Electronics
FOT	Flight Operations Team
GSFC	Goddard Space Flight Center
ISA	Initial Sun Acquisition
LGA	Low Gain Antenna
SA	Solar Array
SAS	Sun Acquisition Sensor
SOHO	Solar and Heliospheric Observatory

GETTING BACK TELEMETRY

Assessment of SOHO status at the loss of telemetry

Owing to Single Event Upsets a built-in protection had switched OFF 3 out of 4 Battery Discharge Regulators (BDR), therefore only one battery was connected to the power bus, which limited the spacecraft autonomy in the case of loss of sun pointing. Without enough power available, the equipment, heaters and instruments were automatically switched OFF. The hydrazine in the tank (about 180 kg) began to cool down and then to freeze slowly.

The spacecraft was already in a "communications backup" configuration: each of the two transponders connected to a Low Gain Antenna: receiver 1 to LGA+Z, receiver 2 to LGA-Z. Since April 1997, receiver 1 is impaired and can be locked only by a slow frequency sweeping. Since this anomaly the lock frequencies of both receivers are not the same.

After the loss of attitude control and telemetry several attempts were made to reestablish communications, including commands to reconnect one battery to the power bus. Owing to the loss of the sun pointing and hence to the onboard power outage, these first sequences were not successful.

At the time of the loss of contact, SOHO was spiraling off the sun with a spin rate (around its X axis, 2000 kgm2 of inertia) estimated to be six to eight degree/second. Flight dynamics analysis and simulation showed that the spacecraft would be spinning around its major inertia axis (Z axis, 3600 kgm2 of inertia) after some time. By the end of June, SOHO would have its solar arrays edge on to the sun. The motion around the sun degree by degree, day after day, increases the period with sunlight impacting on the solar arrays. It means that after 3 months the Z axis would become perpendicular to the sun, which would bring light into the solar arrays and hence power to the spacecraft (see figure 2).

Whether the +Z or the -Z axis would be facing the sun was not known at the beginning of the recovery. On this assumption several thermal simulations were run:

- "Plus Z": +Z axis facing the sun, no power available (hence not possible to get telemetry).
- "Plus Z at 45 degrees": same as above but with +Z axis at 45 degrees of the sun, in such a case there will be some power available, the good receiver (receiver 2) in visibility of the earth and telemetry equipment rather cold (from -10 to -30 C). This might (and in fact did) occur 46 days after ESR-7.
- "Minus Z": -Z axis facing the sun, no power available.
- "Plus Y": Z axis at 90 degrees of the sun which gives full power over half the time.

Overall it appeared possible to reestablish communications with SOHO within six to eight weeks.

According to Flight Dynamics, SOHO would stay on its halo orbit (around L1 point) and diverge only slowly up to mid-November 1998, after that, the orbit would diverge and the spacecraft would have escaped into a solar orbit.

Detection of the downlink carrier

In order to get a "spike" of the downlink carrier, two recovery sequences were defined (new operational procedures). They had to be as short as possible because power was available only for a limited period of time (estimated to be less than 20 seconds). One sequence was to set relays in the Data Handling and for the power supply, which was done by sending decoder commands. The decoders are powered through safe lines as soon as there is power available onboard. The second sequence was to switch ON the data handling and then the transmitter without modulation. These procedures were repeatable and sent either through receiver 1 or through receiver 2 (hence at different frequencies and using several sweeping profiles and rates).

Meanwhile a spectrum analyzer had been installed at DSN station in order to detect any spike of the downlink carrier. Actually many spikes were detected and appeared to be induced by radio interferences, even out of commanding period. Beginning July 1, ESA ground stations, in Australia, Spain, South America and Belgium, reinforced the sky watch to search for the SOHO carrier signal. Spectrum analyzer settings were coordinated between ESA and DSN, measurement results processed and communicated to the FOT at GSFC.

Following a proposition by researchers at the US National Astronomy and Ionosphere Center, on July 23 the 305-m antenna of Arecibo radio telescope (Puerto Rico)was used to perform a bistatic radar test (with a power of 580kW transmitting at 2.380 GHz). This was successful: a 70-m station at Goldstone was able to receive strong echos from SOHO at its predicted location. Moreover the signal width was between 1 to 2 Hz, which is compatible with a spin rate of 53 seconds (determined through Fourier analysis by radar experts at Cornell University). The center frequency drifted slowly by a few Hz indicating a non-principal axis of rotation. Besides an analysis of the collected data indicated a radar cross section of 15 to 20 m*m, compatible with SOHO dimensions. All this gave great hope to recover SOHO.

On August 3 (the 40th day after the loss of contact), the recovery sequence to switch ON the downlink carrier was updated to contain fewer commands (for instance the battery management was taken out) and to add delays between decoder commands. It was sent successfully to receiver 2 (connected to LGA-Z): spikes of the downlink carrier were detected (lasting 2 to 10 seconds, both received by Goldstone and by ESA Perth stations, at 2244.945 Mhz, with a ground AGC of -135 dB).

The carrier was switched ON for increasing period of time (up to 60 seconds). Several times the ground station was able to lock but the duration was still too short to decode the telemetry data. During a test to switch ON transmitter 1, no signal was detected on the ground; which means that the LGA+Z was still not visible from the earth (this fits the case "Plus Z at 45").

Charge of the batteries and first frames of telemetry

Several attempts to charge a battery and to connect it to the bus were not successful. Investigation by battery experts in Europe showed that below 20 Volts, there wouldn't be enough power to maintain the Battery Charge Regulator ON. Therefore to charge one battery it was necessary to keep sending the BCR ON command. On August 8, after 10 hours of such an in-loop commanding, battery 2 was charged up and successfully connected to the bus to get telemetry. To avoid discharging the battery, its two Battery Discharge Regulators (BDR) were opened at the end of the test, which switched OFF the power and hence the transmitter.

These first frames of telemetry confirmed the extreme temperatures, for instance: batteries at -20 C, gyros at about -25 C, some instruments very hot (+80 C) others very cold (-60 C). With the batteries so cold and to avoid any overcharge, the automatic temperature dependent control of the End Of Charge was disabled.

Analysis of the Sun Acquisition Sensor data confirmed a rotation period of 52.6 seconds and that +Z was facing the sun. This also determined the angle from rotation axis to the sun (about 36.7 degrees, measured on August 11).

THAWING THE PROPULSION SUBSYSTEM

Thawing the tank

A new power budget was built based on the power consumption the equipment needed for the coming operations as well as the ratio between charge and discharge of the batteries. Owing to the rotation period, the batteries were in charge only 45% of the time (with a 1A charge current for each battery). This power budget showed that the batteries would charge over several periods if the total power consumption stayed below 67 Watts. In fact to switch telemetry ON consumes 105 W and hence induces a drain of the batteries. Obviously the batteries would have to be charged up each time their voltages will reached a limit, set between 40 and 41 V.

When telemetry was recovered (August 8) the propulsion subsystem was very cold: tank partially frozen (at ~ 1 C), one pipe at -16 C, several thruster as cold as -35 C.
The propulsion experts wisely established that the thawing of the hydrazine must be done first for the tank, then for the pipe section 4 which contains the latch valves. Since latch valve B was open (propulsion branch B used in ESR) it will be closed to cope with any leakage downstream. Then the rest of the pipes sections 1, 3 and then 2 and finally the thruster will be thawed. Doing so will allow any overpressure of thawed hydrazine to flow back to the tank through liquid lines.

On August 12, the nominal and redundant tank heaters were switched ON (total of 32 W). The tank heating was performed with both batteries providing power only to the tank heaters. All the other equipment were switched OFF except for short telemetry checks (temperatures and batteries voltage) every 4 hours. It was necessary to interrupt three times the tank thawing process to recharge the batteries. The total consumed power during heating was about 87W (with telemetry OFF). On the plots of the battery voltages during the whole recovery (in figure 3), one can see the periods of heating and battery charging (which had to be tuned several times after long lively debates between the propulsion/thermal and power experts of the recovery team).

Thawing the tank was achieved (on August 30) after 275 hours of heating (more than 11 days, without taking into account the battery charge periods). It was longer than expected (7 days) owing to higher than estimated heat losses during the interruption of the thawing (to charge the batteries) and also to a more important mass of frozen hydrazine. The tank temperatures given in figure 4 show the slow thawing of the hydrazine.

The "sunheat" mode

It appeared impossible to warm up the whole propulsion subsystem with heaters "digging into" the batteries during each eclipse period. The batteries charge with a 1A current. When telemetry is ON, the power consumption is 105W, which, during each eclipse period, discharge the batteries with a current of 1.25A. To recover SOHO there was a crucial need of switching ON more heaters and equipment without draining the batteries.

Fortunately it was possible to patch the Central On-Board Software (COBS) to use a current like a “fake thermistor” in order to switch ON heaters when power was available from the solar arrays and OFF if not. In a thermostatic regulation, COBS reads a thermistor value and switches a predefined heater OFF when above a "maximum" threshold and ON when below a "minimum" threshold. This was inverted and extended at switching ON a heater when a solar array current was above a "maximum" and OFF when below a "minimum".
Such a patch consisted of changing the following tables (no patch of the code):

- the one defining which thermistor to be used for each thermal control circuit,
- the heaters table to allow to switch ON at the same time both the nominal AND the redundant heaters of the same circuit,
- the memory location of the thermistors.

This patch was first tested on August 19. It had to be reloaded after each battery charging period (during charge the on-board computer is OFF). It is also referred as the “sunheat” mode, which allowed increasing the heating power without discharging the batteries.

First the "solar array current" was used like a fake thermistor which worked perfectly but the power experts thought the heating power would be reduced due to a shadowing effect of the spacecraft body on one of the two solar arrays (this was called the "shunt" mode). Then the charge current of one battery was used instead (referred as "charge" mode). On the other hand as soon as they was enough power available the battery charge current increased from 0 to 1A, therefore the sunheat was active as long as the battery was in charge, sometimes even longer which did drain the batteries a bit. Moreover it was not possible to switch ON more than 150W in "sunheat" mode because this would at once discharge the batteries and hence switch OFF the heaters in "sunheat" mode. These two "sunheat" profiles are given on the top part of figure 5, over the non-eclipse time of each period (when solar arrays in sunlight).

On September 4 a mixed solution was selected:

- the most important heaters for warming up the propulsion were put in "charge" mode (using the battery charge current for the "sunheat" mode) to ensure an equivalent 40% duty cycle of heating;
- other heaters were switched ON few seconds later and switched OFF earlier this was done by using again the solar array current like a fake thermistor ("shunt" mode).

Since this strategy worked well to heat up without discharging the batteries, it was used for more heaters. Overall it consisted of defining a heating power profile that will always be within the amount available from the solar arrays. At the end 48 heaters were put in "sunheat" mode for a total of 517W of heating power. See the "final sunheat" profile on figure 5.

Thawing pipes and heating thrusters

The pipe section 4 was the first one to be thawed, it contains the latch valves and pressure transducers. On figure 4 one can see the increase of temperature and pressure in pipe 4 during its thawing. Pipe 4 was considered thawed on Sept 1; then Latch Valve B was closed (see the increase of branch B pressure up to the back pressure relief level of the valve). Then pipes 1, 3 and 2 thawing were completed on Sept 3.

The propulsion subsystem was kept warm by heating/charging cycles, this done until the final attitude recovery maneuver. Nevertheless it was believed that the thrusters in the spacecraft shadow were not completely thawed (thruster 7B and 8B).

ATTITUDE RECOVERY MANEUVER

Four solutions

Several attitude recovery maneuver were studied:

- ESR without roll control, hence without using thrusters 5B, 6B, 7B and 8B (this solution was the one finally selected mainly because thrusters 7B and 8B were still cold before the maneuver);
- Full ESR recovery which would have used all the thrusters of propulsion branch B;
- Dual spin (in which the spacecraft would have been stabilized around its minimum moment of inertia);
- ISA (Initial Sun Acquisition) recovery, in fact this approach would have used propulsion branch A (in case B side not available).

The maneuver

The recovery maneuver was executed on September 16 with the following steps:

- full battery charge and then a 6-hour heating boost of the propulsion subsystem,
- upload of new onboard standard monitoring to be used to trigger ESR when spacecraft pointing towards the sun,
- test of thruster of branch B, fired for 3 seconds (except 7B and 8B),
- calibration and despin with thruster 4B (down to 0.86 deg/s),
- ESR triggered and sun reacquisition achieved,
- Go back to normal settings of the thermal control.

The gyros were not used for the partial ESR. In fact the roll rate was corrected using thruster 5B or 6B in open-loop from the ground. In ESR, the roll rate was less than 0.2 degree/s, measured through gyro B output and also through analysis of SAS 1B data (a Lissajou plot of pitch and yaw angles gives the sign and the magnitude of the roll rate).

Spacecraft and instruments status at the end of the recovery

Owing to the extreme cold conditions two gyros appeared to be not usable (gyros A and C). Except for these two gyros, all the spacecraft equipment began to work again correctly.

The instruments were checked out between October 5 and 24, remarkably all performed as well as before the mission interruption, some even better.

CONCLUSION AND EPILOGUE

More than 160 members of the SOHO recovery team (ESA, MMS, NASA, ATSC) have performed outstanding work and found imaginative solutions.
On December 21 1998, the third and last gyro was lost, which put SOHO into ESR mode.
With new solutions to:

- reduce the orbit perturbation of the ESR mode (by manual yaw braking from the ground),
- accurately measure the roll rate (wheels spun-up so that there is a net momentum on pitch, which combined with the roll rate creates a yaw disturbance torque),
- patch the Attitude Control Software to fly SOHO without gyro control,
- implement new or updated procedures, including the one to recover from ESR (now done without gyroscope control),

SOHO spacecraft was put back into normal mode on February 1, 1999.
Since then it is the first three-axis-stabilized ESA spacecraft to be operated without a gyro.

REFERENCES

R1 SOHO Mission interruption joint NASA/ESA Investigation board
R2 The SOHO Recovery, on the web at:
http://sohowww.nascom.nasa.gov/operations/Recovery/

TABLES

Events

Date	Time UT	Events
25-Jun-98	04:38	EMERGENCY SUN REACQUISITION (ESR-7)
25-Jun-98	04:43	Interruption of Mission
23-Jul-98	10:00	Confirmation of Orbit Position and Spacecraft Spin Rate by Arecibo and DSN Radar
3-Aug-98	22:51	Reception of Spacecraft Carrier Signal by DSN
8-Aug-98	23:14	Successful battery charge Reception of Spacecraft Telemetry
12-Aug-98	23:39	Begin Thawing of Hydrazine Tank
28-Aug-98	23:02	End thawing of Hydrazine Tank (degrees)
30-Aug-98		Begin Thawing of Hydrazine Lines
16-Sep-98	05:45	Start of Attitude Recovery
16-Sep-98	18:29	ESR 8
16-Sep-98	18:30	SOHO lock to Sun
21-Sep-98	16:58	SOHO in RMW
25-Sep-98	17:30	Orbit Correction (first segment)
25-Sep-98	19:52	SOHO in Normal Mode
Oct 5 to 23		Instruments re-commissioning

FIGURES

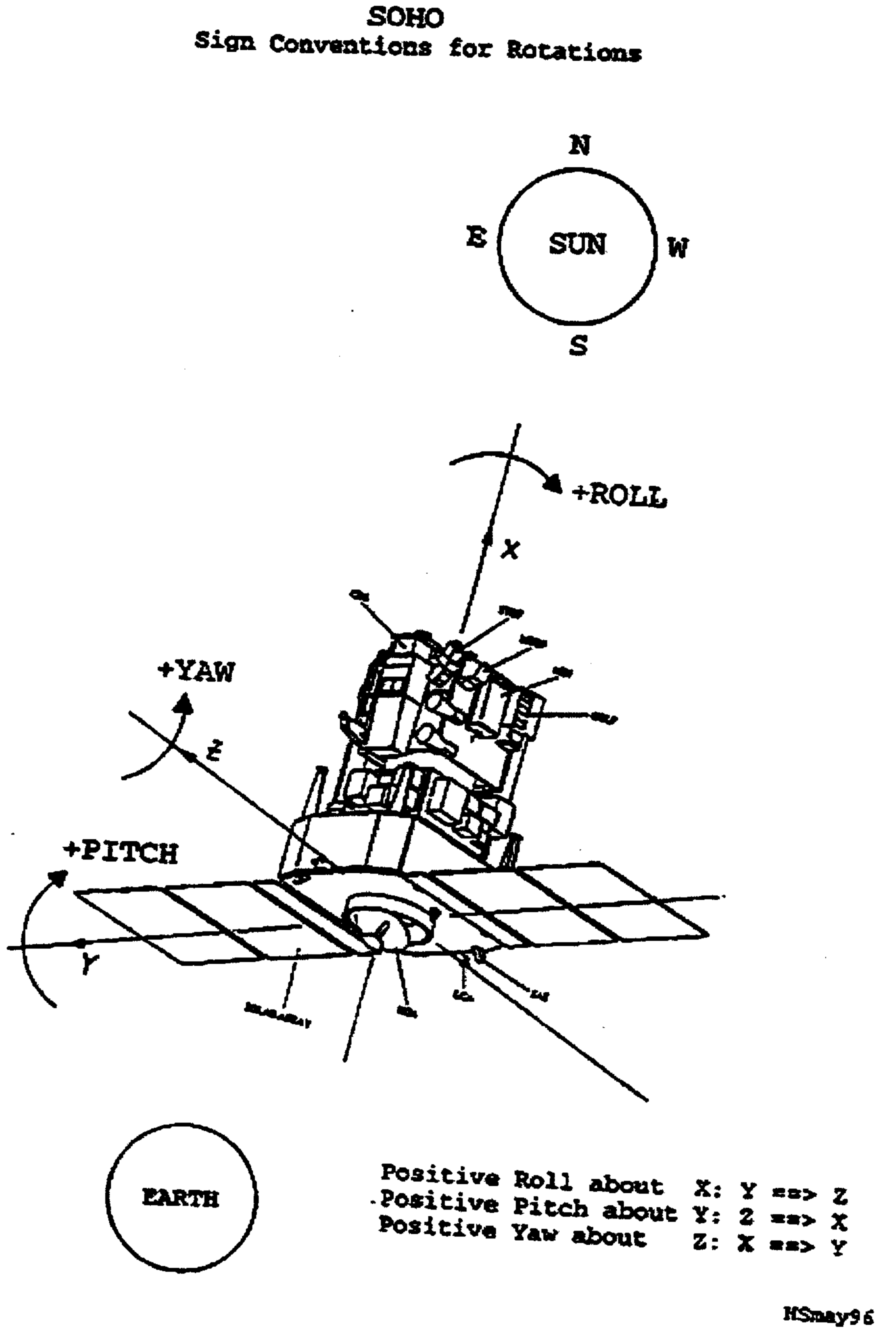

SOHO
Sign Conventions for Rotations
N
E
SUN
W
S
+ROLL
X
+YAW
Z
+PITCH
Y
EARTH
Positive Roll about X: Y ==> Z
Positive Pitch about Y: Z ==> X
Positive Yaw about Z: X ==> Y
HSmay96

Figure 1: SOHO overview and axis conventions

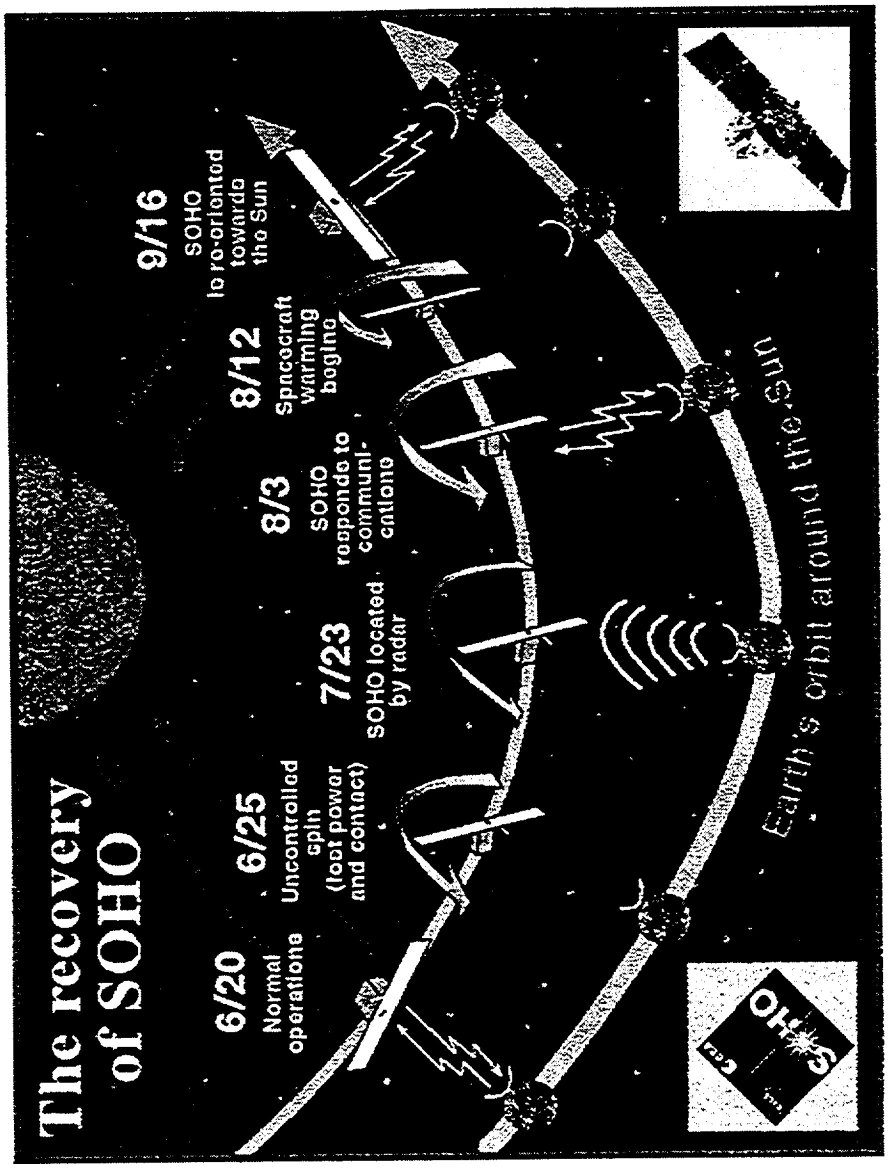
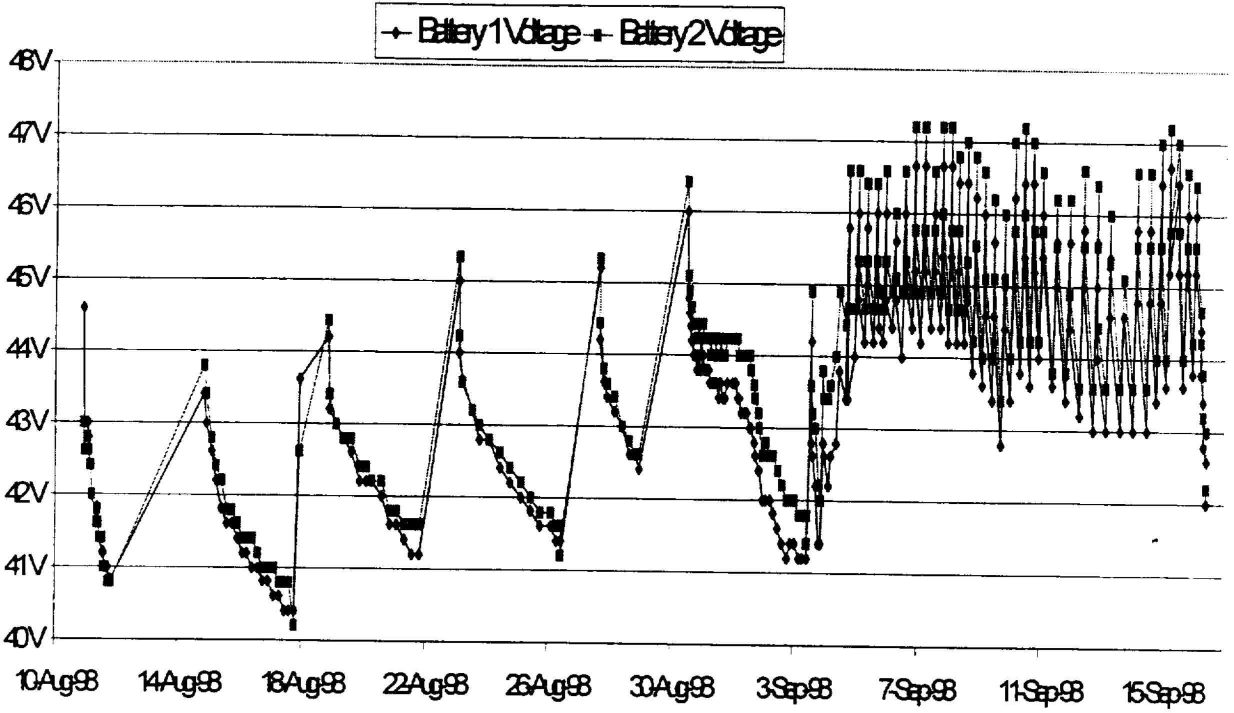
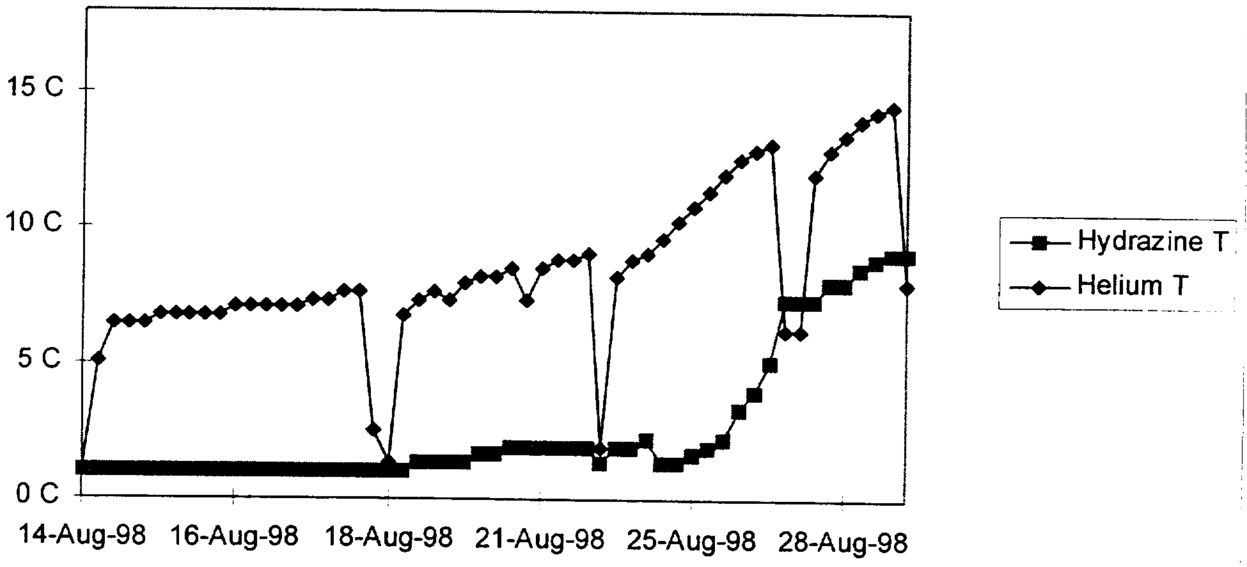

The recovery of SOHO
6/20
Normal operations
6/25
Uncontrolled spin (lost power and contact)
7/23
SOHO located by radar
8/3
SOHO responds to communications
8/12
Spacecraft warming begins
9/16
SOHO is re-oriented towards the Sun
Earth's orbit around the Sun

Figure 2: SOHO recovery phases

Figure3



Tank thawing (figure 4)



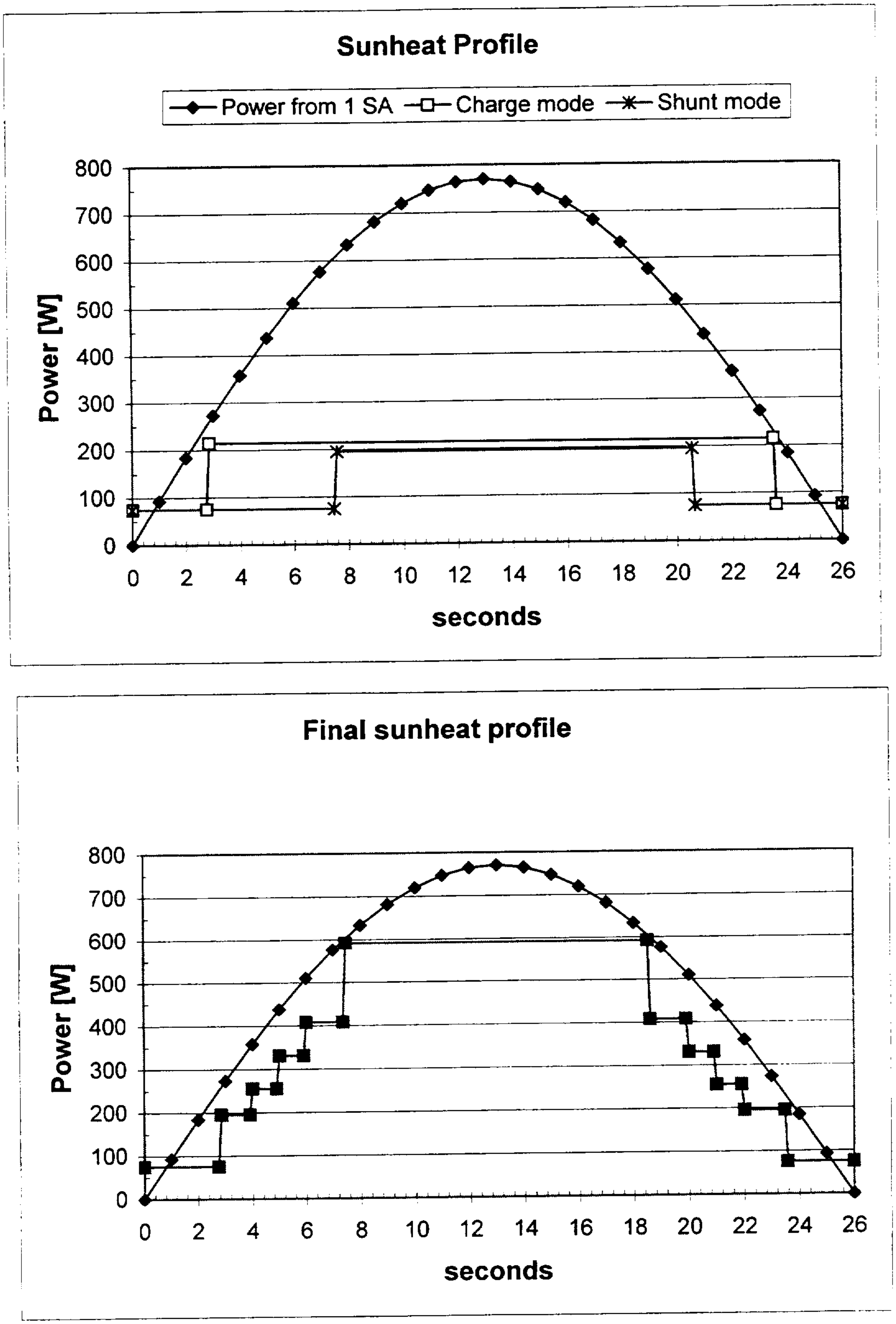
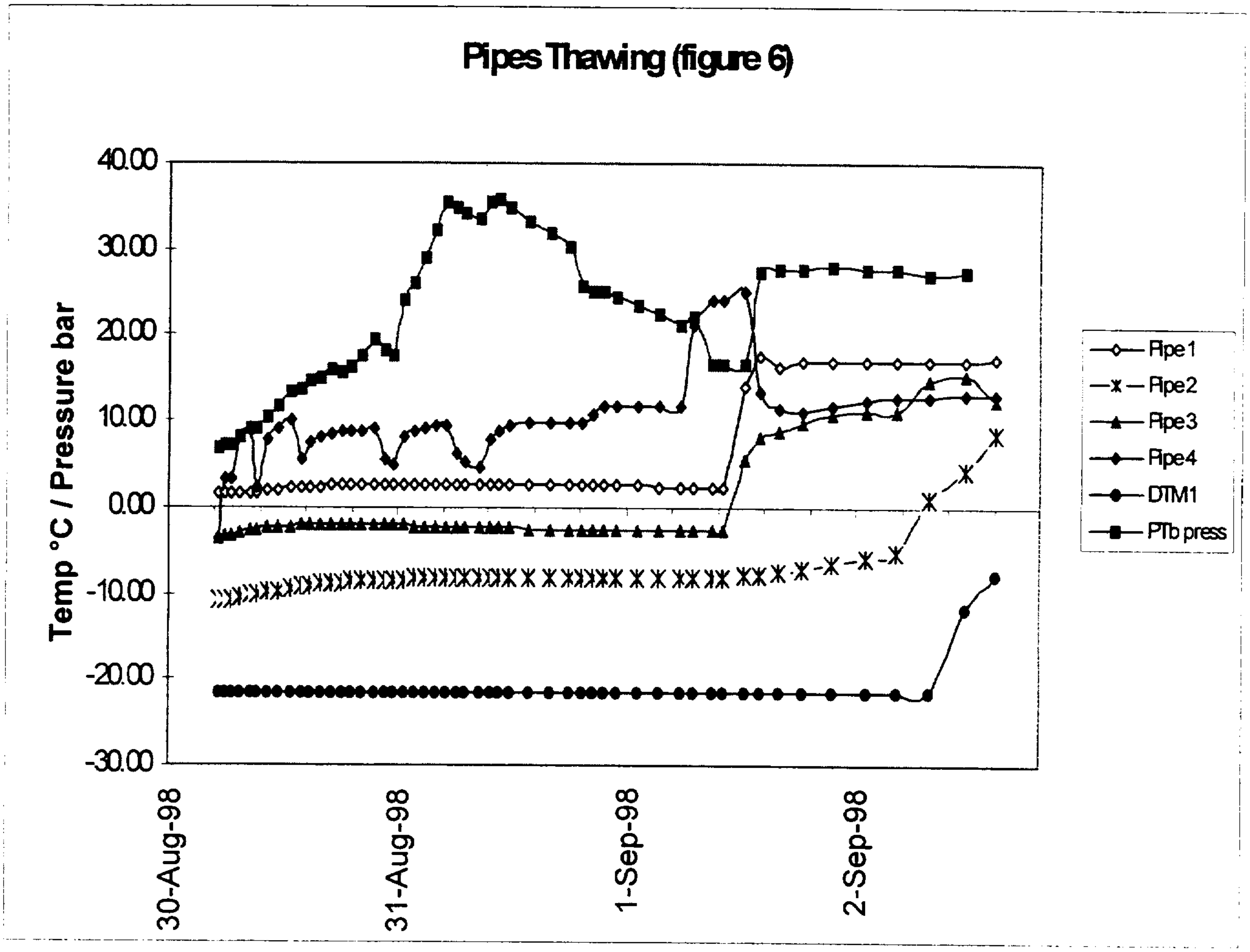


Figure 5: "Sunheat" profiles when solar arrays in sunlight


Pipes Thawing (figure 6)
40.00
30.00
20.00
10.00
0.00
-10.00
-20.00
-30.00
Temp °C / Pressure bar
30-Aug-98
31-Aug-98
1-Sep-98
2-Sep-98
Pipe1
Pipe2
Pipe3
Pipe4
DTM1
PTb press

ATTITUDE SENSOR BIAS EVALUATION FOR ORBVIEW-2

Steve Bilanow and Fred Patt
SAIC General Sciences Corporation

Susan Kennison
OSC OrbImage

ABSTRACT

Postlaunch analysis of onboard attitude measurements and sensor biases from the OrbView-2 spacecraft is presented. With typical sensor measurements combined in single-frame attitude solutions, this spacecraft's nearly subsolar flight path creates very high sensitivity to sensor biases especially around the subsolar point. The time history of the attitude in passage of the subsolar point gives a characteristic signature depending on the sensor errors present. Our tuning of the biases makes use of this subsolar error signature and aims to minimize bias errors particularly at this sensitive point in the orbit. Initially, Sun sensor relative misalignments were manifested in the attitude history by discontinuities at acquisition and loss of individual Sun sensor coverage, and alignment adjustments removed the discontinuities. Magnetometer errors, which contributed to large yaw errors over the night portion of the orbit, were adjusted, and some influence on errors in daylight is noted. Flight data is presented, and various sensor anomalies are noted.

INTRODUCTION

The OrbView-2 (OV-2) spacecraft (originally called SeaStar) carries a single imaging instrument—the Sea-viewing Wide Field-of-view Sensor (SeaWiFS)—which takes multispectral Earth images in the visible and near-infrared spectrum. The spacecraft was assembled and launched by Orbital Sciences Corporation (OSC), which operates the spacecraft through its OrbImage subsidiary, and sells the SeaWiFS data to commercial users and to NASA. The SeaWiFS Project at Goddard Space Flight Center (GSFC) operates a research data system to process, calibrate, validate, archive and distribute data on global ocean bio-optical properties. Primary scientific objectives include quantifying the distribution of phytoplankton and clarifying the ocean's role in the global carbon cycle.

OV-2 was launched in August 1997 from an extended OSC Pegasus vehicle and maneuvered to a 705 kilometer altitude, 98 degree inclination, Sun-synchronous orbit with a local noon descending node. Figure 1 shows the geometry for the routine data collection period for SeaWiFS. Science data is collected onboard OV-2 on the sunlit side of the orbit between +/- 73 degrees of the subsolar point. The SeaWiFS instrument is tilted 20 degrees to reduce Sun glint. Near the subsolar point, the tilt is shifted from aft to forward, and the position of the tilt change is staggered so that global geographic coverage is compiled every two days.

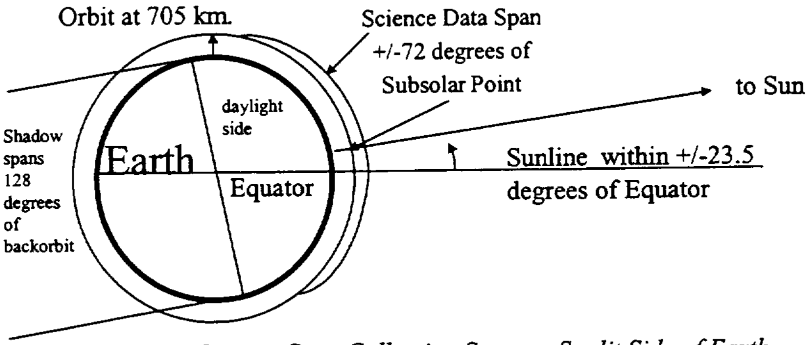


Figure 1. Science Data Collection Span on Sunlit Side of Earth

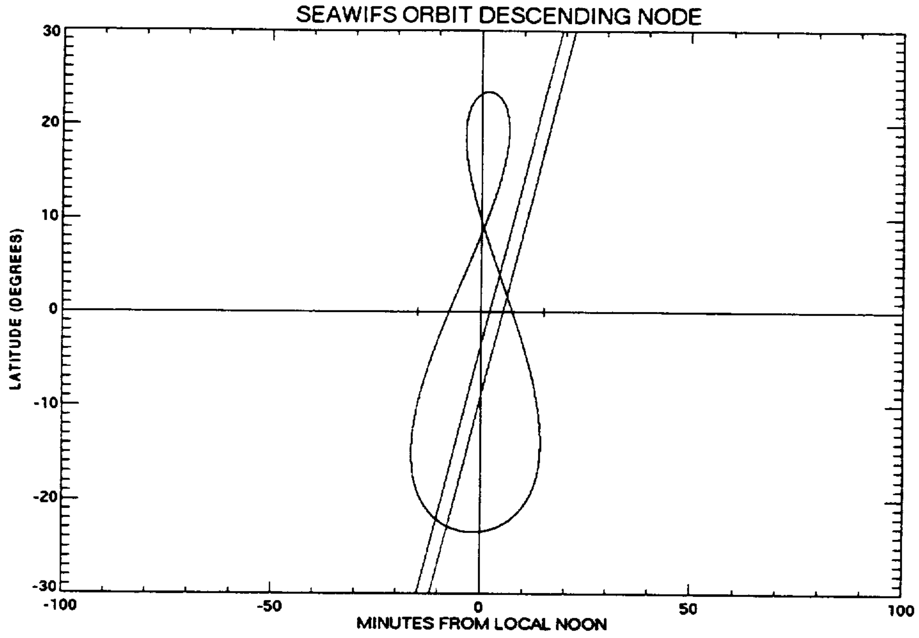


Figure 2. Orbit Plane Relative to the Path of the Sun at Noon

Details of the Sun position relative to the orbit plane are illustrated in Figure 2. This figure shows the position of the OV-2 orbit plane relative to the analemma -- the track of the Sun over the Earth's equator at noon. The x-axis gives minutes from UT noon for local solar noon (with the Earth rotation rate, 4 minutes per degree, giving the scaling to angles), and the y-axis gives degrees latitude. The orbit plane has drifted slowly westward relative to noon since launch, and the diagonal lines show the orbit plane after launch, and on February 10, 1999. The Sun elevation angle from the orbit plane (often called the beta angle) has stayed under 8 degrees throughout the mission thus far. Twice per year the Sun elevation has crossed zero, but most of the year the Sun elevation has been about -3 to -4 degrees.

ONBOARD ATTITUDE PROCESSING AND MODELING

OV-2 is a momentum-biased spacecraft. The Attitude Control System (ACS) utilizes redundant pitch axis momentum wheels and torque rod assemblies for control. The wheel momentum provides gyroscope stability for the pitch axis (roll and yaw stability), and the wheel speed is adjusted to control rotations about pitch. Momentum management and roll/yaw adjustments for the spacecraft angular momentum vector are accomplished by the torque rod interaction with the Earth's magnetic field. Further details about the onboard systems are provided in References 1 and 2.

For attitude sensing, the ACS has 3 two-axis Sun sensors, 2 Earth horizon scanners, and 2 three-axis magnetometers. The mounting geometry for the Sun and Earth sensors is illustrated in Figure 3. The Sun sensors have 128 degree by 128 degree fields of view and are spaced around the front, top, and back of the spacecraft. The middle one overlaps 64 degrees of the front and back sensors, and thus provides some redundancy. With this configuration, Sun measurements are always obtained while the spacecraft is in sunlight with a nominal attitude. The horizon scanners have 45 degree scan cones off the left and right sides of the spacecraft, with the spin axes canted down 5 degrees from the nominal horizontal. More details of the sensors are provided in the ensuing discussion.

All of these sensors were utilized during the first 14 months of the mission, but in November 1998 OV-2 was switched to using a single Attitude Control Electronics (ACE) box. For power and lifetime preservation the plan is to save the redundant ACE with its associated sensors as a cold backup. With ACE B off, the spacecraft uses only Sun Sensors A and B, Horizon Scanner A, and Magnetometer A. Several impacts of the switch to a single ACE will be discussed later.

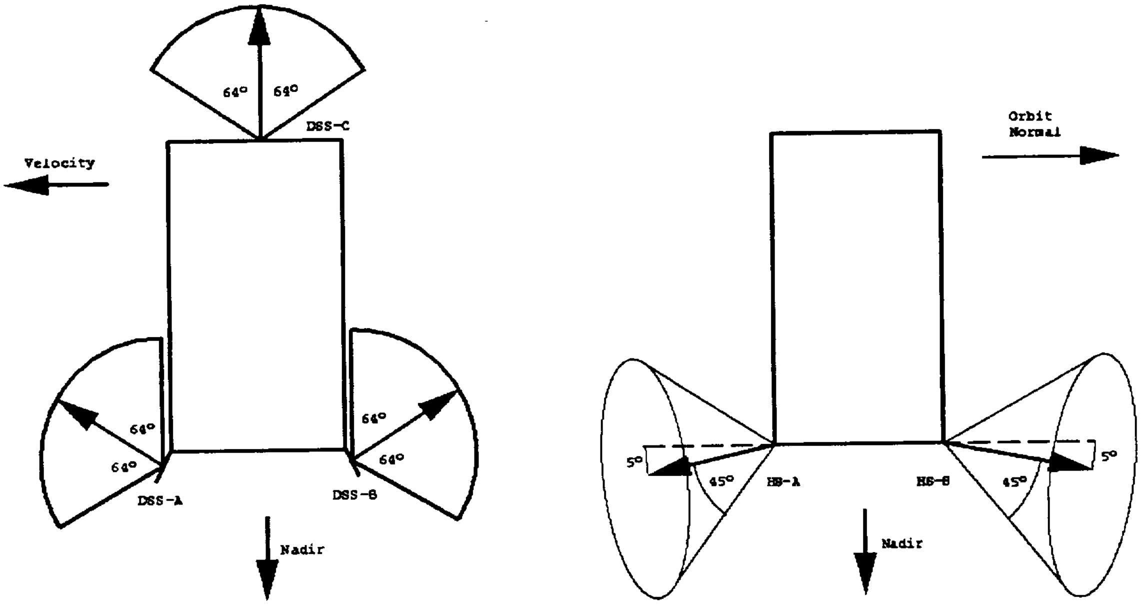


Figure 3. Sun Sensor and Horizon Scanner Mounting Geometry

The ACS uses an implementation of the Quest algorithm (Reference 3) for attitude determination, with sensor inputs converted to vectors and weighted according to nominally expected sensor accuracy. The Sun sensor measurements get the highest weighting, the horizon sensors get the next highest, and the magnetometers get the lowest weighting as summarized in Table 1.

Table 1. Sensor Data Weighting

Sensor	Weighting	Relative Accuracy
Sun Sensor A	10000.	0.01
Sun Sensor B	10000.	0.01
Sun Sensor C	10000.	0.01
Earth Scanner A	50.	0.14
Earth Scanner B	50.	0.14
Magnetometer A	0.1	3.16
Magnetometer B	0.1	3.16

This weighting scheme means that generally the most accurate sensors available dominate in fixing a reference axis in the attitude computation. When two of the same type sensor are available, it is the average of their measurements effectively used. Thus the orbit has two distinct regions:

1) In sunlight, Sun measurements fix one axis of the spacecraft about the reference Sun direction. The rotation of the spacecraft about the sunline is then essentially given by the Earth measurements. (A small contribution from the magnetometers is discussed later.) Since the sunline moves around close to the Spacecraft pitch rotation plane, the Sun dominates in giving pitch information. The Sun also gives yaw information, although around the subsolar point uncertainty in yaw is strongly coupled with uncertainty about the sunline, which depends on the Earth measurements as well. Roll information ends up being provided by the horizon sensors.

2) In Earth shadow, the Earth measurements dominate in fixing the yaw (nadir) axis of the spacecraft, and the rotation about yaw is given by the magnetic field measurement.

When two vectors dominate in determining attitude, there is a well-known problem when these vectors have a very small angular separation. A simple formula for the error E in rotation about the primary reference vector A as a result of position errors in the secondary vector B, can be given as

Rotation error E = (angular error in vector B out of A-B plane) / sin (separation angle)

The errors magnify greatly when the two vectors are very close or nearly opposite, and it was recognized prior to launch for OV-2 that the Sun and Earth vector separation would be a problem. The solution implemented in the onboard software was a "yaw hold" region. Within a table-adjustable tolerance of passing the subsolar point, the yaw value would be held at a constant value. Moreover, a simple lag filter was added to smooth the computed yaw values (using 0.1 at the 2 second control cycle), so it would be a smoothed value held during the subsolar passage. The value selected for the proximity to the subsolar passage was 3.0 degrees. Based on the horizon sensor's specified worst case accuracy of 0.05 degrees, the worst errors just before reaching the yaw hold region could be 0.05/sin (3 deg) = 1.0 degree, which would seem tolerable. However, as postlaunch data proved, the horizon sensor errors could be larger.

An invaluable tool for postlaunch evaluation of the effects of various biases has been a PC-based simulation that included dynamics and environment models and all of the flight software. A large number of parameters used in the onboard code can be adjusted by table uploads, and thus a large variety of biases can be tuned. However, there was not any special software prepared for postlaunch bias calculations and no plan for using certain data in a pre-defined way to adjust particular parameters onboard. Therefore it was necessary to first interpret the flight data to evaluate what adjustments should be investigated further. In many cases, a number of biases could cause the same effects seen in flight, so judgements would be made as to which one to adjust. Then whatever parameters would be planned for adjustment in flight could be readily tested in the simulation, and the simulation proved a reliable testbed. The authors cooperated with OSC engineers in utilizing this tool for analyses (Reference 2).

HORIZON SCANNER BIAS EFFECTS AND ADJUSTMENTS

Following launch, very large errors in yaw showed up around the subsolar point as illustrated in Figure 4. This yaw spike would typically drive maximum torquer activity and generate attitude disturbances as shown. It was easy enough to understand that biases were present. It was perhaps not so easy to understand the time history of the errors, e.g., why was the error much bigger after the subsolar point than before. A simple plane geometry model of the bias effects around the subsolar point was developed to help understand what was going on, and to help guide the selection of the adjustments needed.

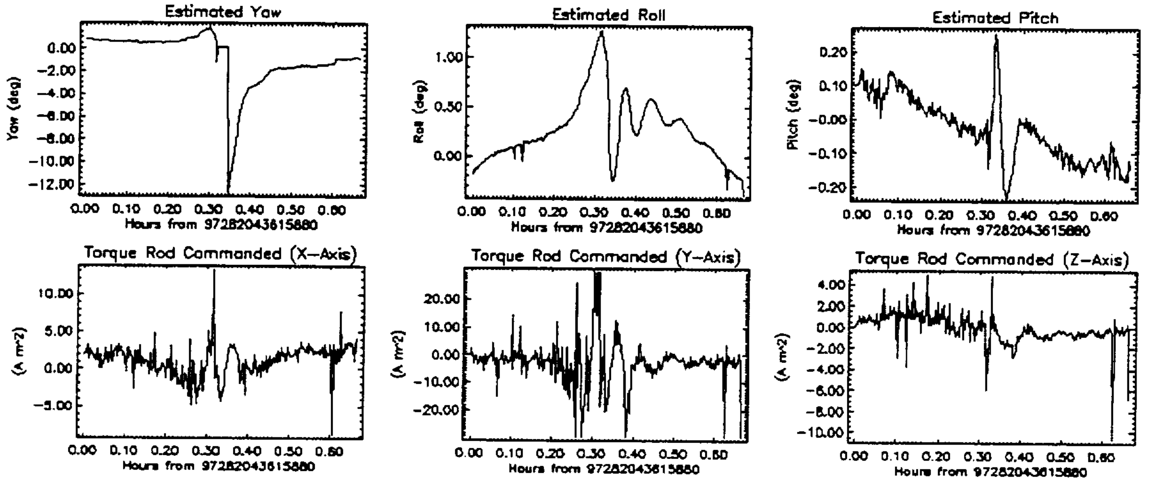


Figure 4. Yaw, Pitch, Roll, and Torquer Activity on October 9, 1997.

A simple schematic of how the bias error causes the yaw errors is illustrated in Figure 5. This illustrates the Sun vector passing close to the zenith vector. One can think of lying on top of the spacecraft looking at the zenith and seeing the Sun go by. But the Earth sensor measures the apparent zenith (opposite the computed Earth vector) as displaced from the true zenith. The yaw error is the rotation from the true Sun-zenith line to the apparent Sun-Zenith line. This error changes rapidly, as shown, as the subsolar point is passed. A spreadsheet implementation of this plane geometry model generated the simulation of the postlaunch biases shown in Figure 5. This model included the yaw smoothing filter and yaw hold.

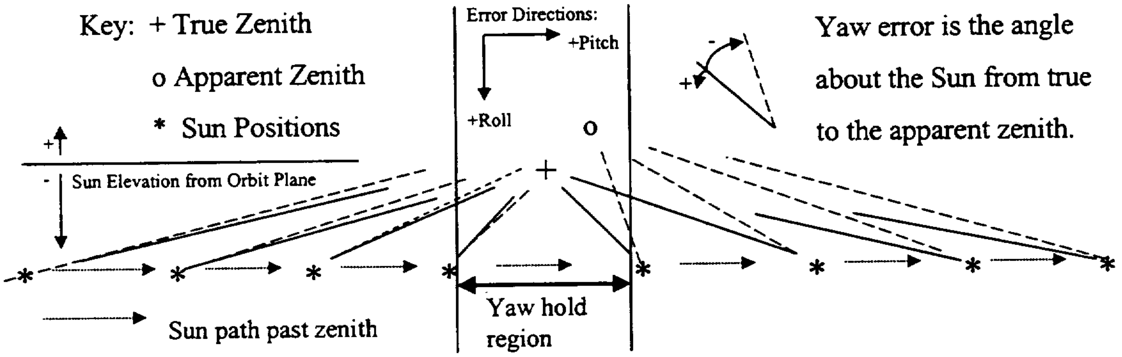


Figure 5. Schematic Diagram of Yaw Errors Generated by Bias

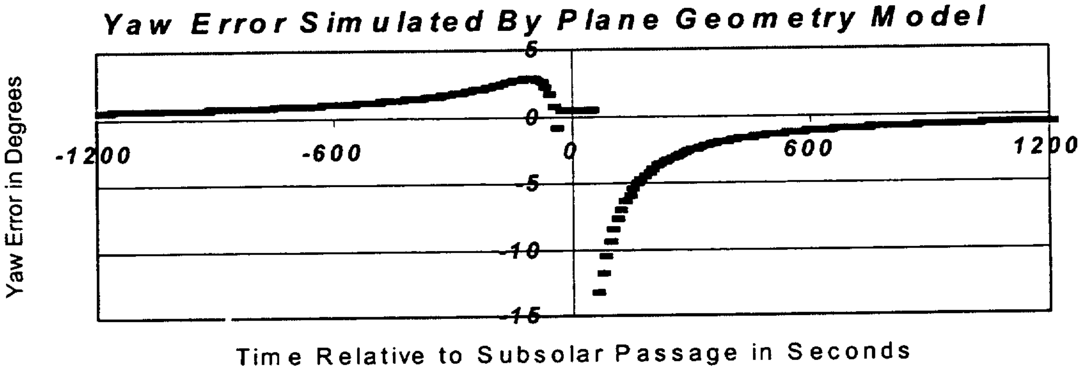


Figure 6. Computed Yaw from Simple Plane Geometry Model of Biases

After the first adjustment was made for Earth scanner biases, the yaw error magnitude near the subsolar point was reduced dramatically, as illustrated in Figure 7 (with note of the scale differences). However, there were still smaller but significant yaw errors present, now with a different time history. Also, the control response was still quite active, and evidence pointed to the roll motion at the mid-point of the orbit being real. The more rapid changes in yaw around the subsolar point, on the other hand, could not be real motion because of the limited control authority of the torque rods.

Reducing unnecessary control activity was an important motivation for further bias adjustments. The poor geometry that causes attitude uncertainty at the subsolar point also contributed to hyperactive, noisy control commands to the torque rods, especially as rate feedback adds to the commands. Unnecessary attitude motion would also make ground computation of the attitude more difficult.

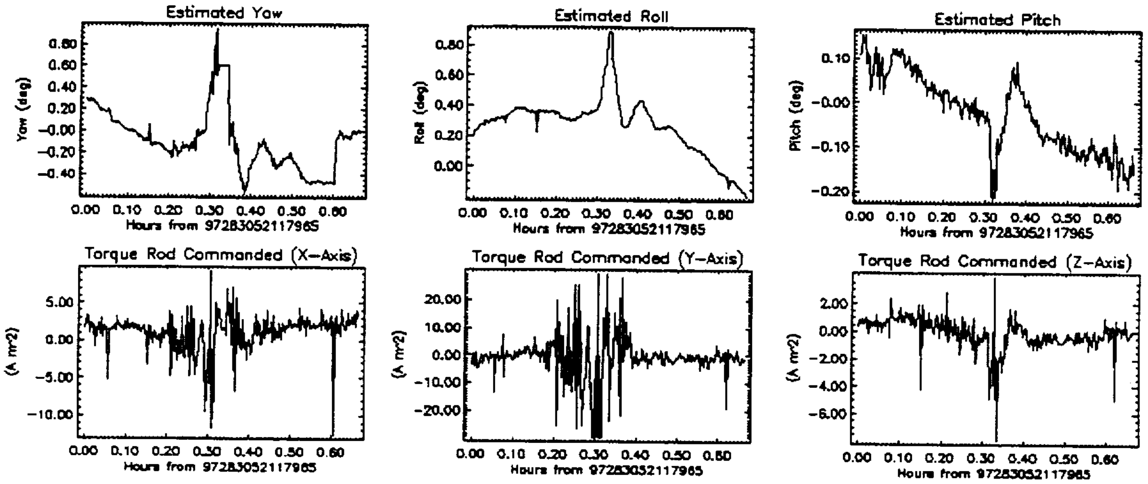


Figure 7. Yaw, Pitch, Roll, and Torquer Activity on October 10, 1997.

Note that noise contributes to some variability in the error pattern each orbit, but general characteristics typically stay the same. Some variation in the pattern has been found to repeat on a daily cycle, and gradual changes in the error characteristics are seen with season, as will be discussed further.

It proved useful, over time, to recognize the effects on yaw error of various bias combinations. Pitch biases tend to cause a yaw error up or down around the subsolar point, while roll biases have effects of different sign before or after the subsolar point. The other important parameters in the plane geometry model are the yaw hold region size and the elevation of the Sun in the orbit plane. The Sun elevation effects are fairly simple to see from the schematic: Pitch error effects tend to diminish and disappear as the Sun elevation approaches zero, while roll error effects get slightly larger. Also pitch effects change sign as the sun elevation crosses zero. The effects for various combinations of pitch and roll biases, and a −3.0 degree Sun elevation, are shown in Figure 8. We refer to the characteristic shapes as the error signatures.

The yaw error amplitude scales linearly with bias amplitudes over our range of tuning, so if the error signature shape is recognized, the bias adjustment amplitude can be readily estimated. Also, using the plane geometry model, one can back out the pitch and roll biases from the error amplitude before and after the yaw hold. This was done to pick bias values to try in the simulations, and worked well initially, but it must be noted that the effectiveness of this broke down at finer levels of tuning. A key deficiency in this approach is that it assumes the biases are the same before and after the subsolar point. This may not be the case due to Sun Sensor misalignments for OV-2, as will be discussed later.

The Earth scanner biases that modeled the initial on-orbit error were 0.75 degrees in pitch and 0.68 degrees in roll. However, different parameters were adjusted onboard. Model parameters that can cause a pitch or roll bias include sensor alignment parameters, assumed scan cone angles, and horizon triggering heights. Also, since it is the average of two Earth scanners that determines the effective Earth vector used onboard, biases could be applied to either sensor or both to achieve the same effect. Double the desired bias must be applied when adjusting only one sensor in order to compensate for sensor averaging. For the first load, just Sensor A was adjusted (somewhat arbitrarily), and the parameters adjusted were to the Earth phase and chord measurements. The phase has nearly a 1 to 1 correspondence with pitch, while the Earth chord changes about 2.2 degrees per degree of roll for the nominal geometry on OV-2. Thus the actual loads were for 3 degrees in Earth chord and 1.5 degrees in Earth phase for Scanner A. The second tweak to the Earth scanners was applied to Scanner B, arbitrarily, and amounted to about 0.05 degree in pitch and 0.16 degrees in roll. A summary of these and further bias loads for the horizon scanners is given in Table 2.

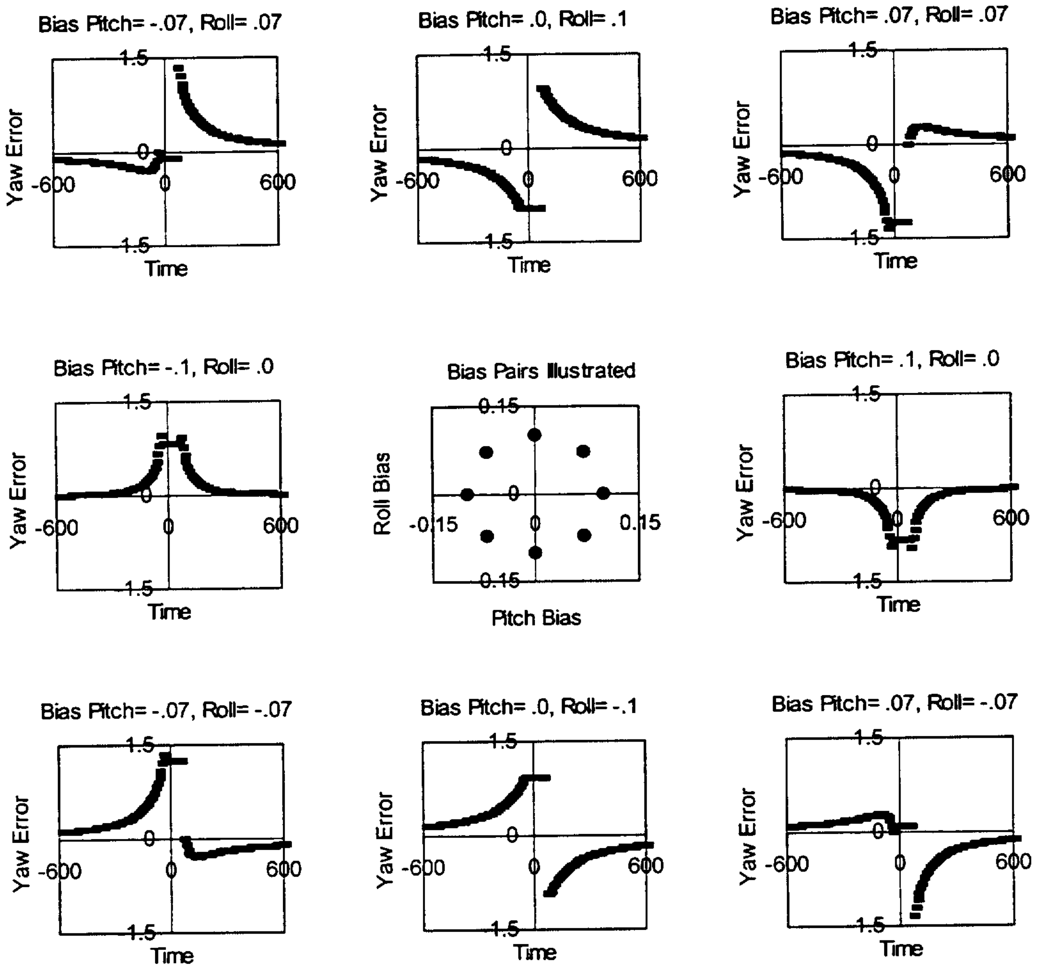


Figure 8. Yaw Error Signatures at Subsolar Passage for Various Pitch and Roll Biases.

Table 2. Horizon Sensor Bias Adjustments Uploaded

Date	HS A Chord	HS A Phase	HS B Chord	HS B Phase	Notes
10/10/97	-3.0	-1.5			Initial adjust from zero values
11/19/97			0.7	0.1	First fine tuning
12/03/97				-0.3	Accompanied Sun Sensor C pitch alignment adjustment
5/6/98				-0.5	Related to seasonal changes
10/28/98	-0.6		3.1		Fix for valid single scanner chords
10/30/98		-1.8		-0.6	Fix for valid single scanner phases
11/01/98		-1.2		0.0	Readjust for single scanner phases
11/06/98	-0.7				ACE A only tuning
12/16/98		-1.3			Further tuning with SSB adjust
03/26/99	-0.6	-1.2			Seasonal adjust and preparation for oblateness re-enabling

The next adjustment to the horizon scanner was needed merely to go along with a Sun sensor adjustment. We have been presenting the error effects so far assuming the Sun sensor is correct and the horizon sensor is biased, whereas actually it is the relative bias between them that makes the difference in creating an error signature at the subsolar point. As our simulator reminded us, when we wanted to adjust Sun sensor C alignment about the pitch axis by 0.2 degrees, we needed to adjust the horizon sensors for a corresponding pitch bias.

With the passage of time it was found that the biases needed adjustment with season. The largest part of this adjustment could be attributed to Earth oblateness effects along with the fact that the subsolar point moves 23.5 degrees north and south of the equator. Earth oblateness is modeled onboard, but the correction is part of a lower priority background task that was disabled early in the mission due to an onboard software problem. This allowed for a slow drift in the bias at the subsolar point. The computed effect is only about +/- 0.1 degrees in pitch at the subsolar point, but with the sensitivity to biases already noted, this could cause yaw errors greater than 1 degree.

Bias adjustments were needed before OV-2 switched to using the single ACE and Earth scanner A only. Whereas we had tuned the biases so that their average vector was accurate, now we had to make sure that each scanner would give a good Earth vector on its own. Several issues had to be dealt with in this process, which we referred to as "balancing" the Earth sensor biases for single string operations.

One problem was with visibility into what the attitude results would be with each sensor used separately. With both sensors on, we only saw their total result, and simulations assured us the same result could be obtained with a whole range of pairs of values. The onboard system had several complex steps that were quite different from the ground data processing stream, so they could not be directly compared. We did not have tools to feed the raw measurements through the onboard simulation, and simulate single sensor results. However, we realized how we could use two derived parameters included in telemetry: the local nadir roll and local nadir pitch. These were generated to provide Earth-scanner-only derived pitch and roll measures to be used in the coarse pointing and despin control modes. The documentation for these parameters stated that they would be based on the dual scanner solution when both scanners were available, but we discovered the code was implemented so that if scanner A was available, it was used first alone. This gave us a straightforward way to tune the proper biases for scanner A, and by maintaining the overall bias balance, also get the correct tuning for scanner B.

Another concern in selecting bias values was that Earth oblateness has a more significant effect on roll when using a single scanner. The Earth radius variation effects are balanced out when two oppositely mounted scanners are combined. Since the oblateness model was disabled, the single scanner roll errors would each have an error that changed at twice the orbit frequency. Thus, the roll measurement from either single scanner could not be right all the time, but the obvious choice for OV-2 was to have the error be as small as possible particularly at the subsolar point. We chose our biases accordingly. The error in the single scanner onboard roll primarily due to oblateness is illustrated by a comparison of the onboard roll and the ground computed roll, as shown in Figure 9. (We should note that the ground data processing model includes some extra oblateness effects to account for systematic Earth radiance variations.) This also shows how the onboard biases were tuned to have a minimum error around the subsolar point.

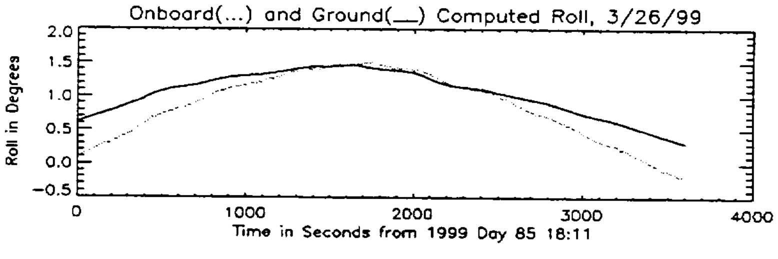


Figure 9. Onboard and Ground Roll with Oblateness Model Differences

One difficulty in comparing onboard and ground results directly comes from differences in handling the manufacturers calibrations for each of the horizon scanners which are implemented in table form onboard the spacecraft. Figure 10 shows one of these as a correction table. There are four such tables onboard, one for the leading edge and one for the trailing edge of each of the two horizon scanners. For ground processing at the SeaWiFS project, it was decided to fit these with a polynomial. It is hard to know whether the wiggles in the curve are an artifact of the calibration process, or something real in the sensor. Since the Earth phase is generally very steady, and the Earth chord varies over a limited range, the sensor generally operates within one wave on Figure 10. Moreover, it is usually at the same place on the curve at the same place each orbit, making the curve's impact hard to separate from other orbit frequency effects.

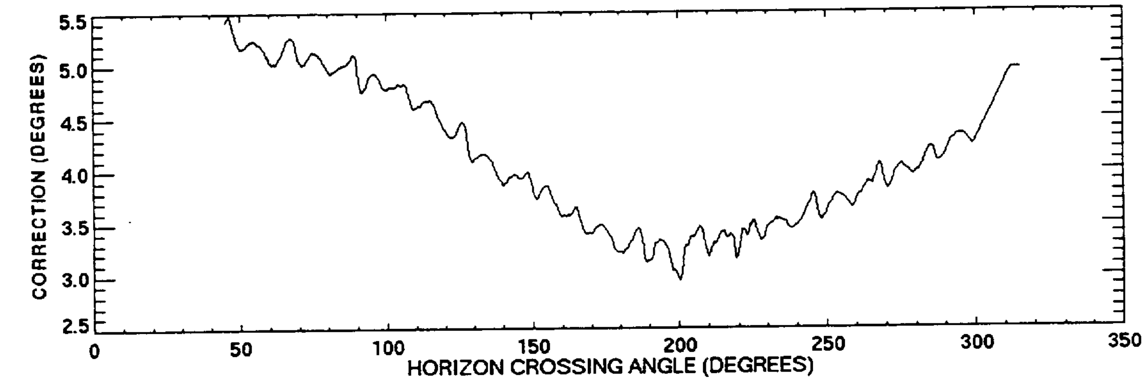


Figure 10. Correction from Calibration Table for HS-A Leading Edge.

A possible effect from the errors in this calibration curve is to cause some of the variability in the subsolar error signature throughout the day. The spin axis attitude currently seems to repeat an average trajectory over the course of the day that varies within 1 to 2 degrees from orbit normal. This is a relatively small change in the operating point on the curve, but it is estimated to that this could have a few hundreths of a degree effect on the roll, and thus a few tenths of a degree effect on the subsolar yaw signature. (More of the subsolar point error variability is now thought to come from magnetometer errors, as discussed later.)

Examples of the variability in the subsolar yaw error signature are provided in Figure 11, which shows 7 consecutive orbits on December 19, 1998. This data was taken just after the switch to the single ACE and the subsequent tuning of biases. There is a predominant underlying pattern to the errors, which was later determined to be due to relative Sun sensor misalignments and is discussed in the next section. However there is also random noise, and a daily recurring pattern that was recognized. The yaw tends to be lower around the subsolar point between about 12:00 to 14:00 UT, and higher around 15:00 to 16:00 UT.

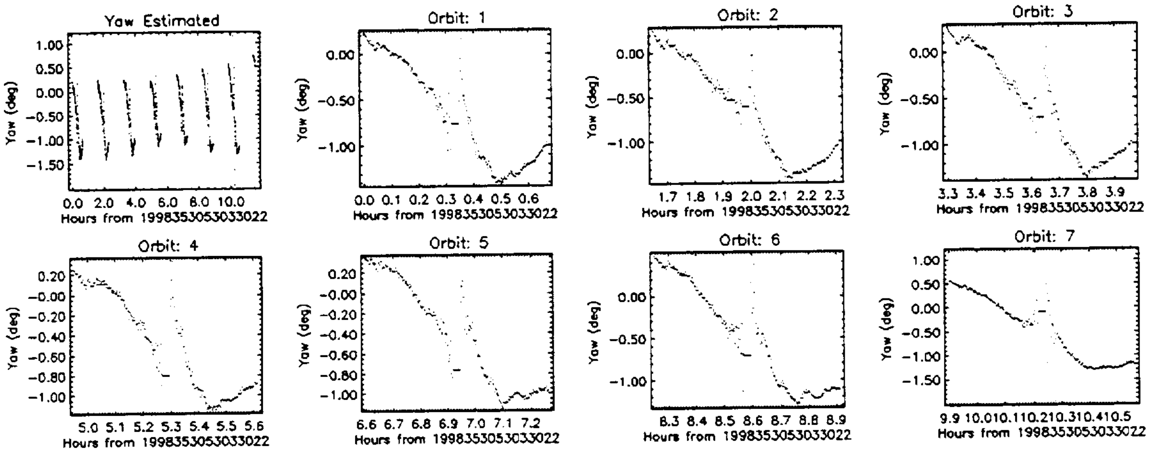


Figure 11. Sample Variability of the Yaw Error Signature around the Subsolar Point.

SUN SENSOR BIAS EFFECTS AND ADJUSMENTS

A helpful way to view the Sun sensor fields of view overlap is to lay them out flat, unwrapping about the pitch axis in a projection on the sky as shown in Figure 12. For this body-centered view, looking out toward the sky with orbit normal up as in Figure 5, the sun moves from left to right across the middle of each sensor Field-Of View (FOV). We will refer to the Acquisition-Of-Signal (AOS) and Loss-Of-Signal (LOS) for the Sun in each Sensor. The Sun encounters the sensors in the order A-C-B. Sensor C overlaps Sensor A and B for about 64 degrees of Sun vector arc. Though the A and B fields of view would nominally just meet, in practice it was found the gap between them varied a bit with the Sun elevation. That gap and misalignments are exaggerated for the illustration. The sensor measurements were labeled as X and Y directions as shown.

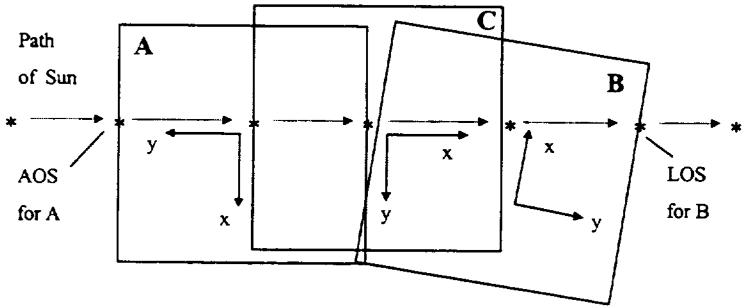


Figure 12. Sun Sensor Field of View Schematic

Just as for the horizon sensors there were notable differences between the onboard and ground processing of the raw sun sensor data. For example, an onboard lookup table was used for each sensor axis based on a manufacturer's calibration. Meanwhile on the ground, a nominal relation for digital sun sensors was used whereby the raw measurements are proportional to the tangent of the angle, and the raw counts were biased or scaled for fine tuning bias adjustments. A sample plot of the difference between the onboard and ground calibration for one of the axes is shown in Figure 13.

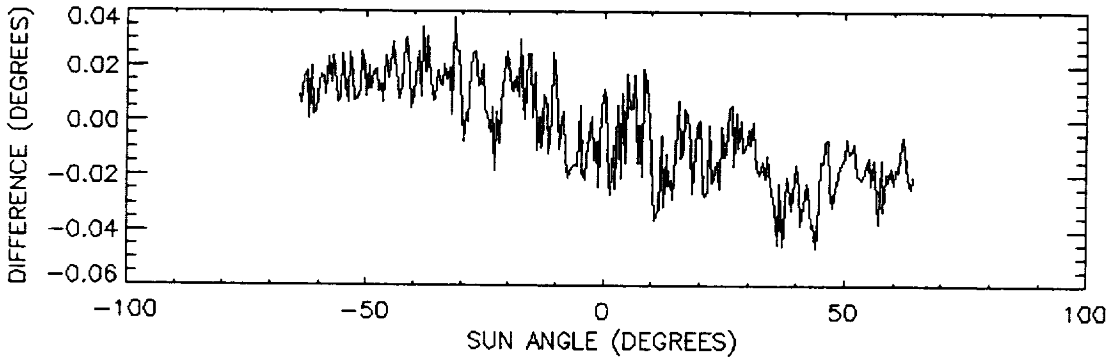


Figure 13. Differences in Onboard Table and Ground Formula for Sun Sensor B Y-axis.

The approach taken for adjusting the onboard alignments was initially a stepwise process where the most obvious errors indicated by discontinuities were progressively removed. This process was partially guided by the SeaWiFS ground processing results, but mainly looked at effects at sensor AOS and LOS. Eventually this process reached diminishing returns as it was apparent that several adjustments could cause the same effect in the flight data. At this point a detailed review comparing the onboard and ground alignments and processing steps provided a finer level of adjustment. A summary of the onboard Sun sensor alignment adjustments is given in Table 3. The last load noted is tentatively planned based on ground alignment results discussed in Reference 4.

Table 3. Sun Sensor Alignment Adjustments

Date	Sun Sensor A Twist, Pitch, Elev.	Sun Sensor C Twist,Elev,Pitch	Sun Sensor B Twist, Pitch, Elev.	Notes (--- indicates no change)
(prelaunch)	.255,.025.,.101	0.0, 0.0, 0.0	-.22, -.134,-.053	Sensor C not measured.
11/8/97			-0.35, ---, +0.7	Yaw adjust of B
12/3/97		----, ----, -0.2		Pitch adjust of C
5/5/98	-0.045, ---, ---		----, +0.066, ---	Twist to A, Pitch to B
12/16/98			----, +0.22, ----	Single String, Pitch to B
3/30/99			-.16,+.30, +.57	Based on Ground Align.
(planned)			----, +.26, ----	Corrected Alignment
(possible)	.032, .18, .049	(n/a)	-.175, .1, .471	Ground Alignments

The need for Sun sensor alignment adjustments was seen clearly right after launch in discontinuities at sensor AOS and LOS. Most dramatically, the jump in yaw at LOS for Sun sensor C can be seen in Figure 7 at 0.6 hours. The goal for initial adjustments was to have the sensors agree at the AOS and LOS points. Fixing errors at these "tie points" would, theoretically, have the sensors be consistent everywhere else. The first adjustment was applied to sensor B, guided by analysis from the ground processing stream that this sensor was more in error. The second adjustment, to Sun sensor C, in pitch, was guided by evidence in the horizon scanners of real pitch changes at the sensor C AOS and LOS.

To help distinguish the sensor AOS/LOS transition effects, especially around the subsolar point where several FOV changes take place in close succession, a FOV limitation was introduced. This also served to remove concerns about the ends of the sensors' calibration curves and any FOV edge effects. The FOV limitation helped separate the errors, but it was still not easy to distinguish AOS/LOS transition jumps from the noise near the subsolar point. For example in Figure 14, there is a jump in Yaw in the May 4 data, just before the subsolar point at Sun sensor A LOS (for the limited FOV) which motivated the Sun Sensor A twist adjustment. However it is hard to see because the subsolar signature is dominated by a horizon sensor pitch bias. Jumps and recoveries in the pitch shown for May 4 in Figure 14 motivated the Sun sensor B pitch axis adjustment, but obviously they are picked out of a lot of noise.

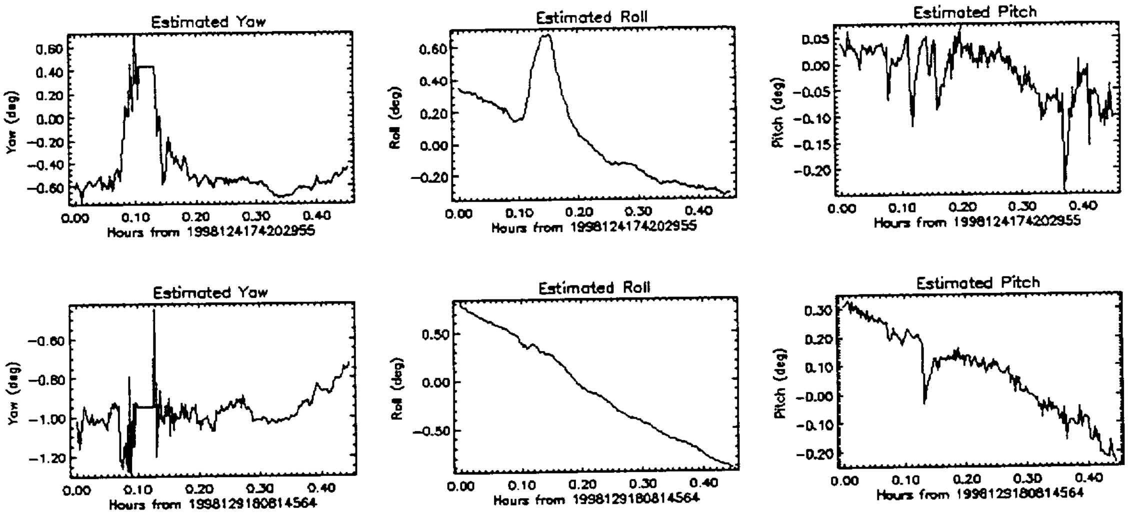


Figure 14. Effects of Bias Changes between May 4 and May 9, 1998.

The effects of several onboard changes are illustrated together in Figure 14. In addition to the Sun sensor alignment adjustments on May 5, there was the horizon scanner B adjustment (from Table 2) on May 6, and a reduction in the feedback control to the torque rods on May 7. The horizon scanner bias changed reduced the bump in yaw that was a characteristic pitch error signature. The roll bump was real motion driven primarily by rate feedback from the yaw error bump, and the overactive torquers. The reduction in torquer commands, by a factor of 0.25, provided the most dramatic change to the attitude system performance. Short term roll and yaw motion was reduced along with the torque levels. At the same time the overall roll yaw amplitude grew as the momentum axis drifted about 1 to 2 degrees from orbit normal and lagged the orbit precession in inertial space. This was judged a very beneficial tradeoff overall, because it made the SeaWiFS pointing changes easier to model in the ground processing software. The remaining bump in the pitch estimate on May 9 deserves a special note: This jump and recovery in the pitch angle is real, and is associated with the change in tilt of the SeaWiFS sensor from aft to fore, which occurs every orbit.

The need for the next tweak to Sun sensor biases became clear after Sun sensor C was turned off. Any bias effects would have been cut in half by the presence of Sun sensor C, averaging the results for the observed Sun direction through the subsolar point (during Sensor A LOS and Sensor B AOS). With C off, pitch motion at Sensor B AOS was observable, and a pitch alignment correction to adjust it was estimated.

After this last bias adjustment was done, it was realized that there was still a residual signature in yaw just after the subsolar point, as is illustrated in Figure 11. It looked like the right half of the signature from Figure 8 for either a -0.1 pitch bias or a +0.1 roll bias. It was observed that this error could be accounted for by a small adjustment in either the twist or the elevation of Sun sensor B. There was no way to distinguish. (The yaw signature could also be simulated by a pitch bias, but that would also show up as real motion via the horizon scanners.) A different approach was taken for selecting further refinements.

The final set of alignment uploads reported in Table 3 was based on the ground alignments translated to give an adjustment to sensor B relative to A. The effects were simulated and seemed to indicate about the correct error (although there was some reservation noted about the pitch change magnitude). The relative error was translated to provide just a correction to sensor B. It was gratifying to note that the alignments accumulated onboard by various small delta corrections had converged close to the one determined on the ground using island targets (Reference 4). These loads were recently sent, and worked as expected, except for a small residual pitch error. This final correction is planned for load soon.

Also, further adjustments to the both of the onboard Sun Sensors currently in use will be considered to make them agree with ground alignments. This last step may not be critical, however, since only the relative alignment errors (which cause biases near the subsolar point) have a significant effect onboard.

MAGNETOMETER BIAS EFFECTS AND ADJUSTMENTS

The Earth and magnetic field geometry has sensitivity to biases near the magnetic poles, similar to the way the Sun/Earth geometry has sensitivity to biases at the subsolar point. Over the magnetic pole, the field lines are along the Earth/nadir direction. For OV-2, magnetometer data is relatively lightly weighted during daylight, but magnetometer bias effects become important in Earth shadow, especially close to the poles. This problem is of secondary concern since SeaWiFS' science data is well inside the daylight portion of the orbit. However, large yaw errors even on the night side are undesirable because they can contribute to attitude disturbances.

The Earth magnetic poles are displaced enough from the geographic pole (about 10 degrees in the northern hemisphere and 15 degrees in the southern hemisphere) that they will pass under the 98.2 degree inclination orbit of OV-2. An overflight of the magnetic pole is not of concern though unless OV-2 is inside the Earth shadow at the time. OV-2 reaches the Earth shadow 26 degrees past the terminator (which is 90 degrees from the subsolar point). With the Sun up to 23.5 degrees from the equator at the solstices, the shadow entry can be as close as 2.5 degrees from the northernmost or southernmost point in the orbit. Thus, OV-2 can be over the magnetic poles while in Earth shadow at the extreme seasons in the winter hemisphere.

The effects of magnetometer biases became quite apparent as the June 1998 solstice was approached, as illustrated in Figure 15. This includes data from the backorbit, outside the science data collection period (see Figure 1). Data is only sampled onboard every 3 minutes during the backorbit period to save onboard storage for science data. Thus the worst case of the yaw errors is probably not shown.

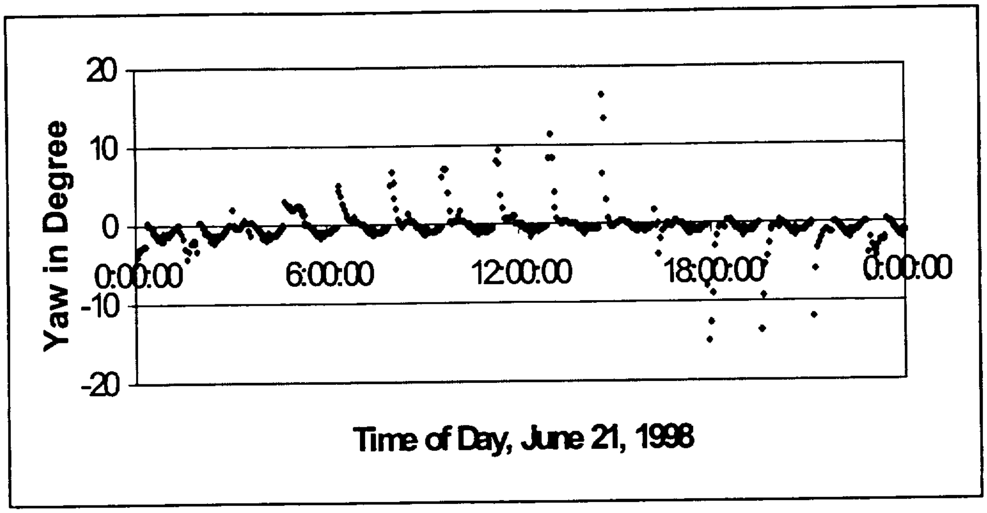


Figure 15. Yaw Errors at Night Due to Magnetometer Biases

The peak yaw spikes seen in Figure 15 occur at the first data sample inside Earth shadow. This is where the spacecraft must rely on Earth and magnetic field vectors for attitude, and vector separation geometry is bad while the spacecraft is closest to the south magnetic pole. The amplitude pattern for the yaw errors each orbit throughout the day can be understood from the effects of the Earth's rotation on the relative Earth/magnetic field geometry. The magnetic pole circles so it is closer and further from the spacecraft's shadow entry point, and the magnetic field vector separation angle from the zenith gets correspondingly smaller and larger. The switch in sign of the error between certain orbits is a similar effect to that of a roll bias when passing the subsolar point (see Figure 8). In this case a magnetometer bias which is along the component of the spacecraft velocity causes a yaw error of opposite sign when the magnetic field direction at the shadow entry point changes as the magnetic pole rotates past.

The magnetometer bias adjustments uploaded for OV-2 are summarized in Table 4. These biases were applied to the zero order term in the polynomial calibration used onboard. (Another bias available for upload only allowed the same bias in all axes due to an oversight.) The first load was done soon after launch to fix a large inconsistency between the Y components of magnetometers A and B. The further biases selected were based on simulation trials to fit the observed yaw error effect. It was clear from analysis of the magnetometer data that there are scale factor and higher order errors present, and/or misalignment errors, but consistent results could not be easily obtained. Thus the tuning was essentially minimizing the errors at the most sensitive place in the orbit (similar to the Earth scanner biases being tuned to the subsolar point in the absence of oblateness modeling). After these magnetometer biases were uploaded it was realized that seasonal effects were changing the yaw errors. After the switch to the single ACE, using Magnetometer A only, an additional tuning was found advisable to reduce yaw spikes during the northern hemisphere winter season. It is clear that further magnetometer calibration would be desirable.

Table 4. Magnetometer Bias Adjustments

Date	Mag A X axis (Y ACS)	Max A Y-axis (-X ACS)	Mag B X-axis (-Y ACS)	Mag B Y-axis (X ACS)	Notes (---- = no change)
Prelaunch	-0.0484	-0.0320	-0.0320	-0.0036	
08/19/97	-0.4684	---------	-0.4520	---------	Made X-axes agree
10/19/98	-0.57	-0.17	-0.35	-0.24	Adjust for yaw spikes
12/15/98	---------	-0.07	---------	---------	Tuning for winter

The effects of magnetometer biases can also be seen on the sunlit side of the orbit. This became especially clear after the switch to the single ACE. In January of 1999 as the Sun elevation reached a maximum, a gap in the Sun sensor coverage between sensor A and B reached a maximum. During this month the gap extended outside the yaw hold region. Thus, this provided visibility of Earth/magnetic field yaw solution for a brief period in the middle of the day. At a certain time of day, every day, the yaw error would be a few degrees negative, and then an orbit or two later it would be over a degree positive. This is illustrated for 3 consecutive orbits in Figure 16.

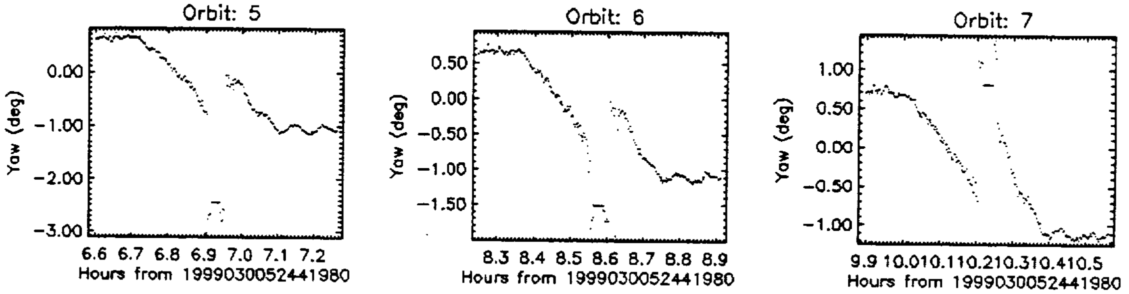


Figure 16. Earth/Magnetometer Yaw solutions during Sun Coverage Gap, 1/30/99.

It is interesting to note that this time of day with the biggest errors corresponds to the South Atlantic Anomaly passage, where the magnetic field is weakest. Constant biases can be expected to have their biggest angular effect when the field magnitude is smallest, so that may contribute to the error here. However the fact that the errors change so much over a small region may point toward another cause: The onboard code uses a 1980 IGRF magnetic field model, and the angular errors in B-field vectors from the current field may be greatest in this region. This merits further investigation.

Whatever the cause, this data highlights an important exception to the rule stated earlier about the Sun and Earth dominating in the yaw estimate. When the Sun/Earth geometry degrades, the magnetic field data comes into play noticeably via the Quest algorithm. In fact, the amount that the magnetic field information comes into play influencing the yaw can be shown to be consistent with the relative weights used in this case. The assumed sensor accuracy is the inverse of the square root of the weight used, and so is about .14 and 3.2 for the Earth and magnetic field respectively. From geometrical considerations already discussed the accuracy is reduced by the sine of the separation angle in computing the rotation direction to another vector. Thus with the Earth within 5 degrees of the Sun, the accuracy provided in yaw by the Earth vector is .14/sin (5 deg.) = 1.6. This implies the Earth vector only provides about twice the accuracy of the magnetic field in providing yaw information very close to the subsolar point. Thus magnetometer errors can noticeably bias the yaw solution at the subsolar point.

Thus we have concluded that some significant amount of the subsolar yaw signature variability comes from the impact of magnetometer errors. The worst yaw error from the magnetometers only occurs at certain times of day, and their effect is reduced by roughly a third of what it would be without Earth data. The amount that the subsolar signature changes at these times of day even with Sun data present (e.g., see figure 11 orbits 5, 6, and 7) is consistent with the data weighting and the errors that come from the magnetometer data (e.g. Figure 16). This suggests a further reduction in the magnetometer weighting may be desirable to reduce yaw errors at the subsolar point.

ANOMALIES

Various glitches—sensor noise spikes and anomalies—encountered are noted here.

- The Sun sensors are prone to bad points at the very edge of the field of view, just before sun presence is lost. (A planned onboard software limit check did not work properly, so a FOV limitation was introduced by special software poke in December 1997 to facilitate the Sun sensor alignment analyses. This poke was not re-introduced after a software reboot in June 1998 because it was judged the FOV edge spikes were tolerable).

- Sun sensor C showed a frequent noise spike at the same place each orbit when the Sun elevation from the orbit plane was about –2 degrees.

- The horizon sensors are prone to small noise spikes, perhaps dozens per day of a degree or two in amplitude. These are suspected to have caused mode changes in two situations where more robust checks could be implemented. A software patch is planned for protection from them during lunar calibration mode by checking for 5 consecutive values before changing modes appropriately.

- Excursions of a couple minutes duration have shown up in the magnetometers, recurring for days at the same place each orbit, with a variable signature as big as 1000 nT. When both magnetometers were on, they showed up simultaneously with the same time history in both.

The causes of these sensor anomalies are not currently understood, but none of them have serious impact on the onboard control, with the impact of transient attitude errors being small. Nevertheless, efforts are being made to obtain more complete statistics on their rates of occurrence for sensor health monitoring.

It is worth noting that data sampling limitations preclude a complete picture of all sensor glitches and anomaly events. The control cycle uses data every two seconds, but we only collect data at this rate during limited real time contacts with the spacecraft and for specially scheduled recorder times. During the normal science data span, data is sampled every ten seconds, and otherwise, during the backorbit, data is sampled only every three minutes. This makes diagnoses of some anomaly events difficult.

Two data anomalies that occasionally have affected the onboard attitude are not due to the sensors:

- In a few cases, the onboard orbit propagator has apparently gone off course somewhat due to poor GPS ephemeris vectors that are apparently not bad quite bad enough to be rejected by certain limits established since launch. After an onboard GPS reset the onboard orbit and attitude estimates recover, but not until after 20 minutes or more of inaccurate attitudes are encountered (especially in yaw around the subsolar point, not surprisingly).

- There is an error of a few seconds in the time tags for all the attitude sensors around the weekly rollover of the GPS time tag counter. This only lasts 10 or 20 seconds, but it does cause a slight pitch disturbance.

These anomalies are under further investigation to see if fixes can be incorporated.

CONCLUSIONS

The postlaunch attitude sensor bias evaluation process for OrbView-2 has been described. Uplink of sensor biases resulting from the analyses have resulted in significant overall improvement in the ACS performance. The effects of various biases, especially near the subsolar point, have been illustrated and discussed. Miscellaneous sensor anomalies and areas for continued work have been noted.

REFERENCES

1. Stolz, P.M., Sivapiragasam S., Anthony, T., "Satellite Orbit-Raising Using LQR Control with Fixed Thrusters," Proceedings of the AAS Guidance, Navigation, and Control Conference, Breckenridge CO, Feb. 1998, Paper No. AAS 98-007.
2. Anthony, T., Sivapiragasam S., "Design and Performance of the OrbView-2 Attitude Control System," 12th AIAA/USU Conference on Small Satellites, Paper No. SSC98-IV-2.
3. Shuster, M.D., Oh, S. D., "Three-Axis Attitude Determination from Vector Observations," Journal of Guidance and Control, Vol. 4, No. 1 pp. 70-77, Jan., Feb. 1981.
4. Patt, F., Bilanow, S, "Attitude Sensor Alignment for Orbview-2 Using Island Targets," Goddard Space Flight Center Flight Mechanics Symposium, May 1999, Paper No. 39.

Attitude Sensor Alignment for OrbView-2 Using Island Targets

Frederick S. Patt
Stephen Bilanow
SAIC General Sciences Corporation

Abstract

The OrbView-2 spacecraft (originally called SeaStar) was launched in August 1997 from a Pegasus vehicle and maneuvered to a 705 kilometer altitude, Sun-synchronous orbit with a local noon descending node. The spacecraft carries the Sea-viewing Wide Field-of-view Sensor (SeaWiFS). The science data processing is performed by the SeaWiFS Project Office at NASA/GSFC. The definitive attitude determination for image geolocation is performed as part of Level 1 product generation by post-processing the spacecraft attitude sensor data. The geolocation accuracy requirement is 1 pixel (2 sigma), which corresponds to 1.12 km at nadir.

Post-launch alignment of the attitude sensors (Sun sensors and horizon scanners) to the SeaWiFS instrument was needed in order to meet the accuracy requirement. This is typically performed in flight by co-aligning the sensors to achieve internal consistency within the ACS, and then computing a fixed ACS-to-instrument transformation. For OrbView-2 this did not produce consistent geolocation results, due largely to seasonal and geometric variability shown by the sensors.

An alternative approach has been developed which uses navigation assessment results derived from the SeaWiFS science data. This technique involves identifying islands in the image data and matching them with reference locations from a catalog, and was implemented to automatically collect navigation error statistics on an ongoing basis. These results have been utilized to analyze systematic attitude errors and compute corrections which are applied to the sensor alignments. This approach has been used to substantially improve the consistency of the overall aspect determination and to evaluate improvements to the processing software. The alignment results have also been utilized to compute uplink corrections for the onboard ACS.

1. Introduction

The Sea-Viewing Wide Field-of-view Sensor (SeaWiFS) is an ocean color sensor which was launched on the OrbView-2 spacecraft in August 1997. The spacecraft (called SeaStar prior to launch) was built and launched by Orbital Sciences Corporation (OSC), who were also responsible for the orbit raising, initial activation and check-out of the spacecraft. After OrbView-2 was declared operational and routine SeaWiFS data collection was started, spacecraft operations responsibility was assumed by OrbImage, a subsidiary of OSC. The SeaWiFS Instrument collects science data both via an onboard recorder, which is dumped twice daily, and through direct broadcast of the data to ground stations.

The SeaWiFS Project Office at NASA Goddard Space Flight Center (GSFC) obtains the SeaWiFS science data in a unique arrangement called a "data buy". In this arrangement, the SeaWiFS Project schedules the SeaWiFS instrument operation and onboard data recording, and also collects direct broadcast data from authorized research stations. NASA is responsible for all research and educational use of the data, while OrbImage retains all commercial data rights.

The SeaWiFS Project processes all of the data and generates data products for use by the Ocean Science community. The data are processed from Level 0 to Levels 1A, 2 and 3, using software developed and operated by the SeaWiFS Project. The data products are transmitted to the GSFC Distributed Active Archive Center (DAAC) for distributed to authorized users. Details of the SeaWiFS data processing, product definitions and project activities can be found on the SeaWiFS home page (http://seawifs.gsfc.nasa.gov).

An integral part of the Level 0-to-1A data processing is the determination of the navigation information, e.g., the spacecraft definitive orbit and attitude, to support the instrument data geolocation. The spacecraft attitude is determined by post-processing of the telemetry from the onboard attitude sensors. The accuracy requirement for geolocation is 1 pixel (~1.6 mrad, or 1.12 km at nadir), 2 sigma.

The assessment of navigation accuracy is performed using the method of island target matching which was developed prior to launch (Patt, Woodward and Gregg, 1997) and refined using the flight data (Patt, Woodward and Gregg, 1997). In order to meet the accuracy requirements, it has been necessary to utilize the results of navigation assessment to determine the co-alignments of the attitude sensors.

The remainder of this section describes the SeaWiFS data collection, the OrbView-2 attitude control system (ACS) and the SeaWiFS Project attitude determination algorithms. Section 2 describes the initial effort to achieving consistent co-alignment of the attitude sensors and the problems encountered, and Section 3 discusses the use of the island target matching for sensor alignment and the results obtained with this method. The spacecraft orbit and sensor geometric characteristics are summarized in Table 1.

Table 1 - Spacecraft Orbit and SeaWiFS Viewing Geometry

<u>Orbit Characteristics:</u>	
Type	Polar, near-circular, Sun-synchronous
Altitude	705 km
Inclination	98.2 degrees
Period	99 minutes
Equator crossing	Local noon (descending)
<u>SeaWiFS Geometry:</u>	
Instantaneous FOV	1.5897 mrad (0.091 degree)
Ground IFOV	1.12 km (at nadir)
Scan half-width	58.3 degrees
Pixels along scan	1285
Scan period	0.167 second
Sensor tilt	+/- 20 degrees
Sensor coverage	40 minutes per orbit centered within the orbit day

1.1 SeaWiFS Data Collection

The SeaWiFS instrument is commanded to collect data for 40 minutes each orbit during the sunlit part of the orbit, in order to observe locations where the solar zenith angle is 72 degrees or less. The instrument is also tilted to avoid Sun glint from the ocean surface, 20 degrees aft prior to the mid-point of orbit day and 20 degrees forward afterward.

The instrument can produce data in two formats: full-resolution, local-area-coverage (LAC data, and 4x4 subsampled, global area coverage (GAC data). The data are collected by two paths. An onboard recorder records data during scheduled instrument operation, and is dumped to a ground station twice daily. Due to storage limitations, the majority of recorded data are in GAC format, with a limited amount of LAC storage available for calibration. Full-resolution data are also direct-broadcast continuously in high-resolution picture transmission (HRPT) format during scheduled instrument operations, and can be received during spacecraft overpasses by stations with SeaWiFS direct broadcast licenses.

The SeaWiFS Project collects and processes both the onboard recorded data and direct-broadcast data from research HRPT stations. A SeaWiFS scene is either a scheduled orbit of GAC data (40 minutes), a scheduled LAC segment or a single HRPT station overpass. Except where noted, the analysis presented here was performed with full-resolution (HRPT and LAC) data. Although these data do not provide full global coverage, the higher resolution has proven to be invaluable in assessing and refining SeaWiFS navigation.

1.2 OrbView-2 Attitude Control System

The OrbView-2 ACS was designed by OSC (Anthony and Sivapiragasam, 1998). The attitude sensor complement consists of three two-axis digital Sun sensors (DSSs), two infrared horizon scanners (HSs) and two three-axis magnetometers. For purposes of definitive attitude determination, the sensors of interest are the DSSs and the HSs.

The attitude control hardware consists of redundant three-axis magnetic torque rod assemblies and pitch-axis momentum wheels. The torque rods are the sole source of roll and yaw control. The momentum wheels are used in concert with the torque rods for pitch control, and also to maintain a pitch momentum bias for attitude stabilization.

The spacecraft coordinate system is defined with the X axis downward, the Y axis opposite the orbit velocity direction and the Z axis toward the orbit normal. Thus, at the nominal spacecraft attitude, the spacecraft X-Y plane corresponds to the orbit plane and the spacecraft Y-Z plane corresponds to the local tangent plane.

The geometry of the three DSSs is illustrated in Figure 1. Each DSS has a field-of-view (FOV) of +/-64 degrees in each axis. All three DSSs are mounted with their boresights in the spacecraft X-Y plane and with one sensor axis parallel to the spacecraft Z axis. DSS-A is canted approximately 64 degrees in the direction of the spacecraft velocity, DSS-B is canted in the opposite direction and DSS-C is oriented toward the zenith. Thus, the three sensors provide overlapping coverage. DSS-A provides coverage from the beginning to approximately the middle of orbit day, DSS-B provides coverage from the middle to the end of orbit day, and DSS-C covers approximately 35 minutes centered within the orbit day. Each sensor provides two digital outputs (one for each axis) with a resolution of 0.0002 (about .011 degree at the boresight), and an associated time tag.

The HS geometry is illustrated in Figure 2. The HS design uses a conical scan with a half-cone angle of 45 degrees. HS-A is oriented with its rotation axis opposite the orbit normal, and HS-B is toward the orbit normal; both HSs have their rotation axes canted 5 degrees downward from the spacecraft plane corresponding to the local tangent plane. Each sensor provides time-tagged Earth width and phase measurements with a resolution of 0.0055 degree.

One aspect of the OrbView-2 ACS configuration is that the choice of orbit (local noon equator crossing) causes the Sun and nadir vectors to be nearly anti-parallel at the mid-point of the orbit day (the sub-solar point). The maximum angle varies seasonally, but remains close to 180 degrees, and twice yearly the two vectors become exactly anti-parallel. Thus the attitude geometry is always poor at the sub-solar point, specifically for the yaw angle; the spacecraft ACS disables yaw control at this part of the orbit.

One final note is that the redundant ACS sensors, DSS-C and HS-B, were powered off in November 1998 to conserve lifetime and electrical power. Although this occurred after the work described here was performed, it was a factor in some aspects of the analysis, as noted below.

1.3 SeaWiFS Project Attitude Determination Algorithms

The following is a summary of the attitude determination algorithms used in the SeaWiFS Project science data processing software. A detailed description is being published as a NASA technical report. As previously stated, this processing is performed as part of the Level 0-to-1A conversion.

The determination of the spacecraft orbit is a precursor to the attitude determination. The orbit determination is based on the output of a GPS receiver carried on the spacecraft. The GPS orbit vectors are fitted to an orbit model which includes a high-order geopotential field model, third-body (Sun and Moon) attraction and atmospheric drag.

The attitude determination processing is performed individually for each SeaWiFS scene. The first step is to unpack the attitude sensor telemetry from the spacecraft data stream and convert the samples to engineering units. The data are validated using absolute limits and sample-to-sample consistency checks. They are then smoothed and interpolated to the SeaWiFS scan times using overlapping cubic polynomials. The measurements are then combined to produce a measured Sun and nadir vector in the spacecraft frame for each scan line. If multiple DSS measurements are available (i.e., where DSS-C coverage overlaps the other sensors) a weighted average is computed. When both HSs are operating a least-squares solution for the nadir is performed; otherwise a deterministic algorithm is used.

The attitude determination is performed using a Kalman smoother, which is a two-pass (forward and reverse) Kalman filter. This algorithm reduces the discontinuities at transition points (i.e., start/end of DSS coverage), stabilizes the yaw angle through the sub-solar point and provides continuity of the attitude determination during periods of missing or invalid attitude sensor data (in particular, the Sun coverage gap at the sub-solar point following the DSS-C power-off). The state consists of the attitude (roll, pitch and yaw) angles, and the state propagation uses a simple dynamics model which assumes an inertial pitch rotation axis (due to the due to the stability provided by the momentum wheels) and a fixed pitch angle. The attitude is propagated to each scan line time and updated using the preprocessed DSS and HS data, with the measurement noise covariances selected to place more weight on the DSSs. The final attitude angles are computed as a weighted average of the two passes, with the weighting determined by the covariance from each pass.

2. Co-alignment of the Attitude Sensors

2.1 Method of Attitude Sensor Co-alignment

The initial methods used to perform co-alignment of the OrbView-2 DSSs and HSs were based on the assumption that consistent attitude determination results could be obtained by analysis of the attitude sensor data alone, which would then require at most a constant transformation from the spacecraft to the SeaWiFS reference frame to produce consistent geolocation. This implies that the attitude sensors, particularly the DSSs, perform consistently over the full range of angles encountered during normal spacecraft operation.

These initial efforts relied heavily on the output of DSS-C, since this was the only sensor whose coverage overlapped with all of the other sensors. The expectation was that, if the other sensors could be aligned consistently with DSS-C, this would produce the required consistency of attitude determination. The approach was to first, compute the alignments of DSS-A and DSS-B with respect to DSS-C; and then, to use the combined output of all three DSSs to compute the alignments of the HSs.

In both cases, the alignments were computed through a least-squares minimization process. For the DSS alignments, since the observed vector (the Sun direction) was the same for all measurements, the method was to minimize the vector differences between the results from each DSS. The calculation was performed separately for DSSs A and C and for DSSs B and C, in the corresponding overlap regions.

The HS alignment to the DSSs was somewhat more complicated, since the geometry of the reference Sun and nadir vectors had to be considered. In this case, the observable quantity was the angular separation of the Sun and nadir vectors. The method was to minimize the differences in the angular separations of the measured and reference vectors in the least-squares sense. It should be noted that the observability of the HS-to-DSS alignments is limited by two factors: first, at the nominal attitude the nadir vector is maintained close to the spacecraft X axis, making this vector largely insensitive to yaw misalignment of the HSs; second, the OrbView-2 orbit geometry, which maintains the Sun vector close to the orbit plane, provides very little visibility on the roll co-alignment, although some knowledge of the roll alignment of the HSs relative to DSS-C can be gained at the sub-solar point.

2.2 Evaluation and Analysis of Alignment Results

The accuracy and consistency of the spacecraft attitude has been assessed throughout the mission using the method of island targets (Patt, Woodward and Gregg, 1997). In this method, the SeaWiFS science data are filtered to classify individual pixels as land, water or clouds. Islands are located in the data as small, contiguous groups of land pixels surrounded by water, and free of clouds. The centroid of each island is computed using the geolocation based on the spacecraft orbit and attitude, and the island locations are matched with reference locations in a catalog. The coordinate differences and correlative data are saved for analysis. This processing is performed on each SeaWiFS scene following Level 0-to-1A conversion.

The co-alignment of the attitude sensors was performed in early November 1997. The results of the island target matching were monitored over the next few months to determine whether the 1-pixel requirement was being met. Typically, 10 days of matching results would be collected and analyzed at a time; this provided significant numbers of observation (~3000) for each period while still maintaining reasonable visibility on temporal changes.

The results of this analysis are shown for two 10-day periods (Figures 3 and 4): days 300 through 309 of 1997, and 30 through 39 of 1998. The matching results, which are initially determined as latitude and longitude errors in degrees, have been converted to along-scan and along-track errors in SeaWiFS pixel units, and plotted vs. latitude (note that at this time HRPT data were only available for the Northern hemisphere. Figure 3 shows that both along-scan and along-track results were consistent, if somewhat noisy, in early November 1997, although there was a significant shift in the along-track errors between about 43 and 52 degrees latitude. By early February 1998 (Figure 4) the results had deteriorated, with more latitudinal variation seen in the along-track errors and more noise along-scan.

The deterioration in geolocation accuracy can also be observed in statistical measures of the results. Table 2 shows the median and inter-quartile range (IQR) of the errors for both periods. (The IQR is used as a measure of the statistical error because it is less sensitive to outliers than the root-mean-square, or RMS error). For a normal distribution, the IQR is equal to 1.35 standard deviations; therefore, an IQR of 0.675 pixel is equivalent to a 2-sigma error of 1 pixel. Although the median errors remained within a reasonable range, the IQRs exceeded this value at the start and increased steadily during this period.

Table 2 - Initial Geolocation Error Results

Period	Coordinate	Median (pixels)	IQR (pixels)
1997300 - 1997309	Along-scan	-0.007	0.827
	Along-track	0.254	0.866
1998030 - 1998039	Along-scan	-0.125	1.083
	Along-track	0.072	0.990

The attitude sensor alignments were re-evaluated in March of 1998 in an attempt to reduce the systematic errors; as before, the results were initially consistent but deteriorated over a few months. To understand the cause of these changes, the results from the individual DSSs were analyzed for consistency. This analysis was continued through June of 1998, to include the full range of seasonal effects.

This analysis was performed as follows. The samples from of each DSS were converted to unit vectors and transformed to the spacecraft frame. This was performed for full GAC scenes of data from several seasons (October and December of 1997, and February, March and June) representing the full range of Sun-to-orbit plane angles. The angle between the Sun vector and the spacecraft X-Y plane was then computed for each vector, and the results plotted for all three DSSs on the same axes (Figure 5). The angles for DSS-C are plotted as a solid line, and those for DSS-A and B (first and second half of the scene, respectively) as points. The angles range from about -4 degrees (in June) to 4.5 degrees (in February).

The plots show that the consistency between DSS-C and the other two sensors varies significantly. In October and June, when the Sun angle was negative, DSS-A and C are reasonably consistent, but the difference between DSS-B and C exceeds 0.2 degree at some points. In December and March, when the Sun angle is small, DSS-B and C are more consistent, but the differences between DSS-A and C are larger, exceeding 0.1 degree at times. This is also observed in February, when the Sun angle is positive. Also, DSS-C shows more noise than the other sensors, with jitter and spikes exceeding 0.1 degree at times.

These results show clearly why it was not possible to obtain consistent co-alignment of the sensors within one SeaWiFS pixel (0.09 degree). For DSS-A and C, the optimum co-alignment differed between October and February by at least 0.1 degree. For DSS-B and C in October and June, the consistency varied by more than 0.2 degree within a single orbit.

It is not possible from this analysis to determine which DSS output is more valid, or would produce the most consistent spacecraft attitude; the only conclusion is that the DSSs themselves are inconsistent at a level which is significant compared to the accuracy requirement. However, based on the plan to power off DSS-C as well as the observed higher noise, it was decided to minimize the use of this sensor, and to attempt alignment of the remaining sensors by analysis of the island matching results.

3. Alignment of the Attitude Sensors Using Island Matching Results

The results of the island target matching consist of the island location errors in latitude and longitude, along with the island coordinates, scan angle (i.e., pixel number), and other information (number of pixels, catalog ID number) for each matched island. Given a set of island matches spanning a range of scan angles, the average spacecraft attitude error can be estimated. Similarly, given a set of attitude error estimates spanning a range of attitude sensor outputs (i.e., Sun vector positions within the DSS FOV), it is possible to estimate the attitude sensor alignment errors.

3.1 Calculation of Partial Derivatives

Estimation of the spacecraft attitude error requires that the sensitivity of the island locations to attitude changes (i.e., partial derivatives of the latitude and longitude with respect to roll, pitch and yaw) be calculated. Although this can be performed analytically, a numerical approach was chosen for simplicity. This was performed in two stages. First, look-up tables were generated, which contained partial derivatives of SeaWiFS observed latitudes and longitudes with respect to roll, pitch and yaw, as a function of observation pixel number and latitude. Then, the latitudes and pixel numbers for a set of island matches were used to interpolate from the table values, to generate the partial derivatives for every island in the set.

The use of look-up tables indexed only by latitude and pixel number is based on the fact that both the partial derivatives and the observed latitude are functions of the same parameters: the orbit latitude and altitude, spacecraft attitude, tilt angle and pixel number. Further, the observed latitude can be used to infer the orbit latitude, if the other quantities are known. By assuming that the orbit altitude, spacecraft attitude and tilt angle are fixed (a reasonable approximation in this case), a fixed relationship is established between observed latitude, orbit latitude and pixel number.

This relationship was used to generate the look-up tables as follows. First, a nominal set of observed locations was generated using a simulated orbit, fixed tilt and nominal attitude. Then the attitude angles were perturbed by small amounts, one at a time, and the observed locations were regenerated. The differences in observed locations, which represent the numerical partial derivatives, were computed and saved for each attitude angle, along with the nominal latitudes and pixel numbers. Since the tilt angle can be either forward or aft, two sets of tables were generated, one for each tilt.

After the tables were generated, the partial derivatives for each island were computed. For each island, the associated pixel number was used to select a table column; the row was selected by finding the tabulated latitudes within that column which bounded the island latitude. The partial derivatives were then interpolated to the island location using the values from the selected column and rows. The output was a set of six partial derivatives, of the island latitude and longitude with respect to spacecraft roll, pitch and yaw.

This approach makes some assumptions, most notably that the actual variations in the spacecraft attitude have a negligible effect on the partial derivatives. While this is not entirely correct, for the typical range of attitude angles (+/-1 degree) the effect on the partial derivatives is a few percent. Since the alignment corrections being estimated are of the order of 0.1 degree, errors of a few percent in the partial derivatives are acceptable.

3.2 Estimation of Alignment Errors and Results

Once the partial derivatives have been computed for a set of island matches, they can be combined with the island location errors to estimate the average attitude error using the least-squares method. However, interpretation of the attitude error in terms of sensor misalignments is not necessarily straightforward, and requires knowledge of how each sensor is affected the spacecraft attitude. The HS data are affected by the roll and pitch, but are mostly insensitive to yaw; the DSS data are affected by all three angles, but while the sensitivity to pitch is constant, the sensitivity to roll and yaw varies with orbit position (i.e., as the Sun vector moves in the spacecraft reference frame). In addition, for SeaWiFS attitude determination, the DSS data are weighted more heavily than the HS. Thus, the pitch and yaw angles depend almost exclusively on the Sun sensor data, while the roll angle depends primarily on the HS data for much of the orbit. Finally, since DSS-A and DSS-B coverage is mutually exclusive, the attitude errors resulting from each can be analyzed independently.

This approach can be illustrated by plotting errors vs. pixel number for two latitude ranges and a given time range. In this case, the island matches for Days 200 through 209 of 1998 were used. The plots (Figure 6) show along-track error vs. pixel number for two latitude ranges: -10 to 0 degrees and 30 to 40 degrees. The first plot shows an overall positive bias and a negative slope, which result from attitude errors of -.05 degree in pitch and .12 degree in yaw, respectively. The second shows a positive slope, resulting from a -.14 degree yaw error. The data in the first plot lie south of the sub-solar point, within the range of DSS-B, while the second plot is north, within DSS-A range, so the corresponding attitude errors can be applied to the alignments for each of these sensors.

The analysis of the island location errors was performed using island matches from 10 days of data at a time, which typically provides 5000 to 10,000 matches. This was repeated for different time ranges spanning months of data to derive the current best set of attitude sensor alignments. The result has been a reduction in the systematic error, so that the 1 pixel geolocation accuracy requirement can be met. These results are shown by plotting island location errors vs. latitude for some recent data (Figure 7). The plots, which contain 6400 island matches, show a significant reduction in systematic errors compared to previous results, although some structure is still evident, particularly near the equator. The IQRs for these data are 0.569 and 0.564 pixel for the along-scan and along-track errors, respectively, well within the requirement of 0.675 pixel.

It must be noted that these results, while encouraging, have not yet been produced with the full range of seasonal variations (i.e., Sun angles) seen during the mission. The navigation assessment results from future processing may indicate a need for further alignment corrections or other refinements to meet the accuracy requirement under all conditions. Also, there are other factors affecting attitude determination which have not been considered here (e.g., seasonal horizon scanner triggering height variations and the limitations of the current attitude dynamics model). Nonetheless, this approach has resulted in significant improvements in SeaWiFS geolocation accuracy. The method of island targets produces large numbers of error measurements and good coverage of the geometric, geographic and temporal range of instrument observations, providing good visibility on systematic errors, as well as the means for developing and evaluating future improvements.

4. Conclusion

A method of determining the OrbView-2 attitude sensor alignments has been developed, using navigation assessment results based on island target matching. This method has been found to produce more consistent attitude determination results than co-alignment of the sensors alone, and as a result the SeaWiFS instrument geolocation accuracy requirement is currently being met. Navigation assessment processing and analysis will be continued to verify future geolocation accuracy.

References

Anthony, T. and Sivapiragasam, S., "Design and Performance of the OrbView-2 Attitude Control System", 12th AIAA/USU Conference on Small Satellites, September 1998.

Patt, F., Woodward, R., and Gregg, W., "An automated method for navigation assessment for Earth survey sensors using island targets", *International Journal of Remote Sensing*, vol. 18, no. 16, 1997

Patt, F., Woodward, R., and Gregg, W., "Automated Navigation Assessment for Earth Survey Sensors using Island Targets", NASA/Goddard Image Registration Workshop, Goddard Space Flight Center, November 1997.

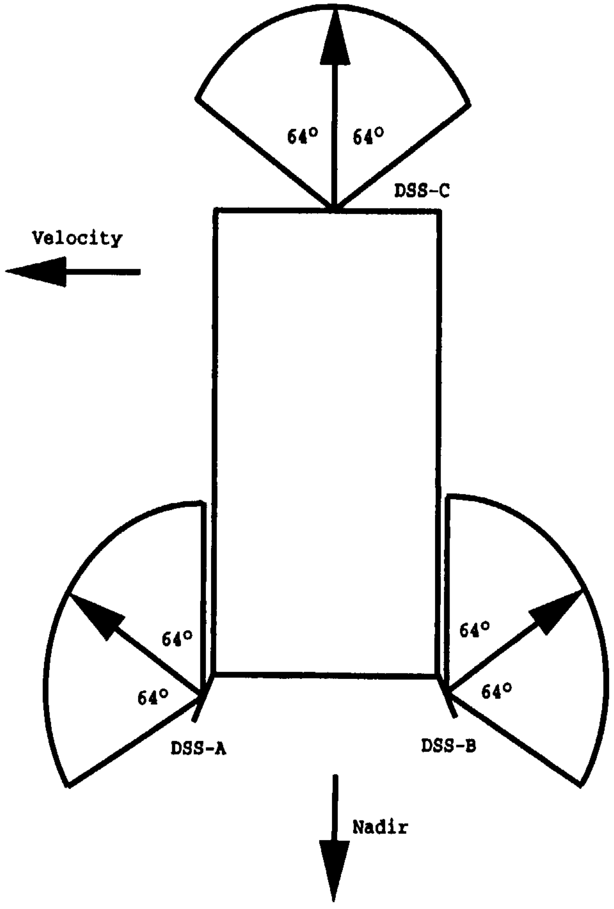


Figure 1 - OrbView-2 Digital Sun Sensor Coverage

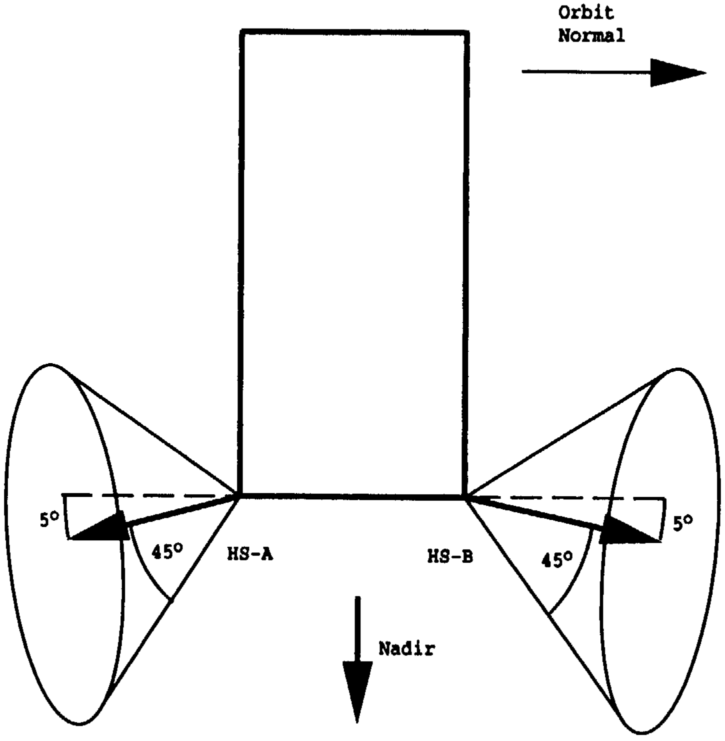


Figure 2 - OrbView-2 Horizon Scanner Geometry

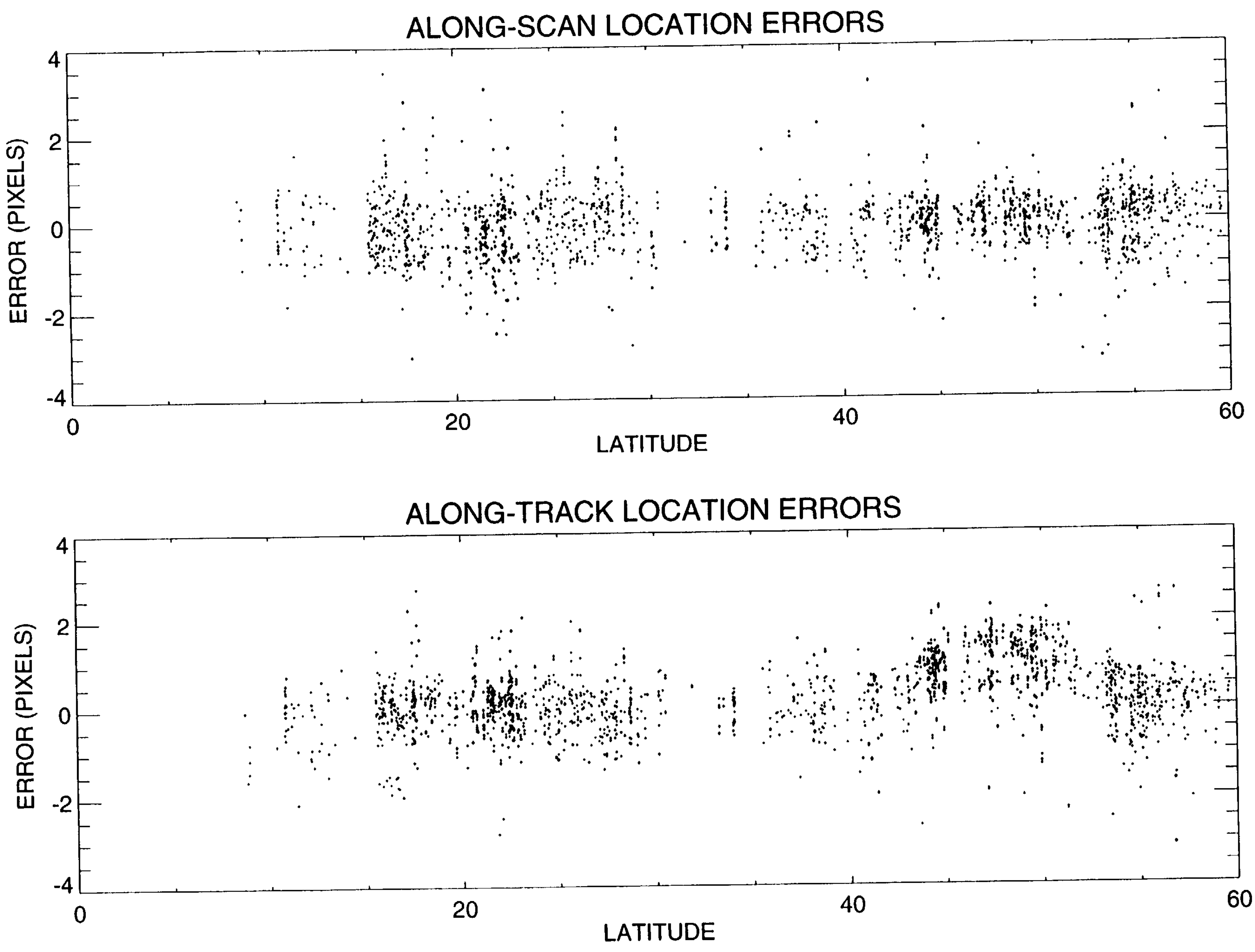


Figure 3 -- Island Location Errors vs. Latitude: Days 300 - 309 of 1997

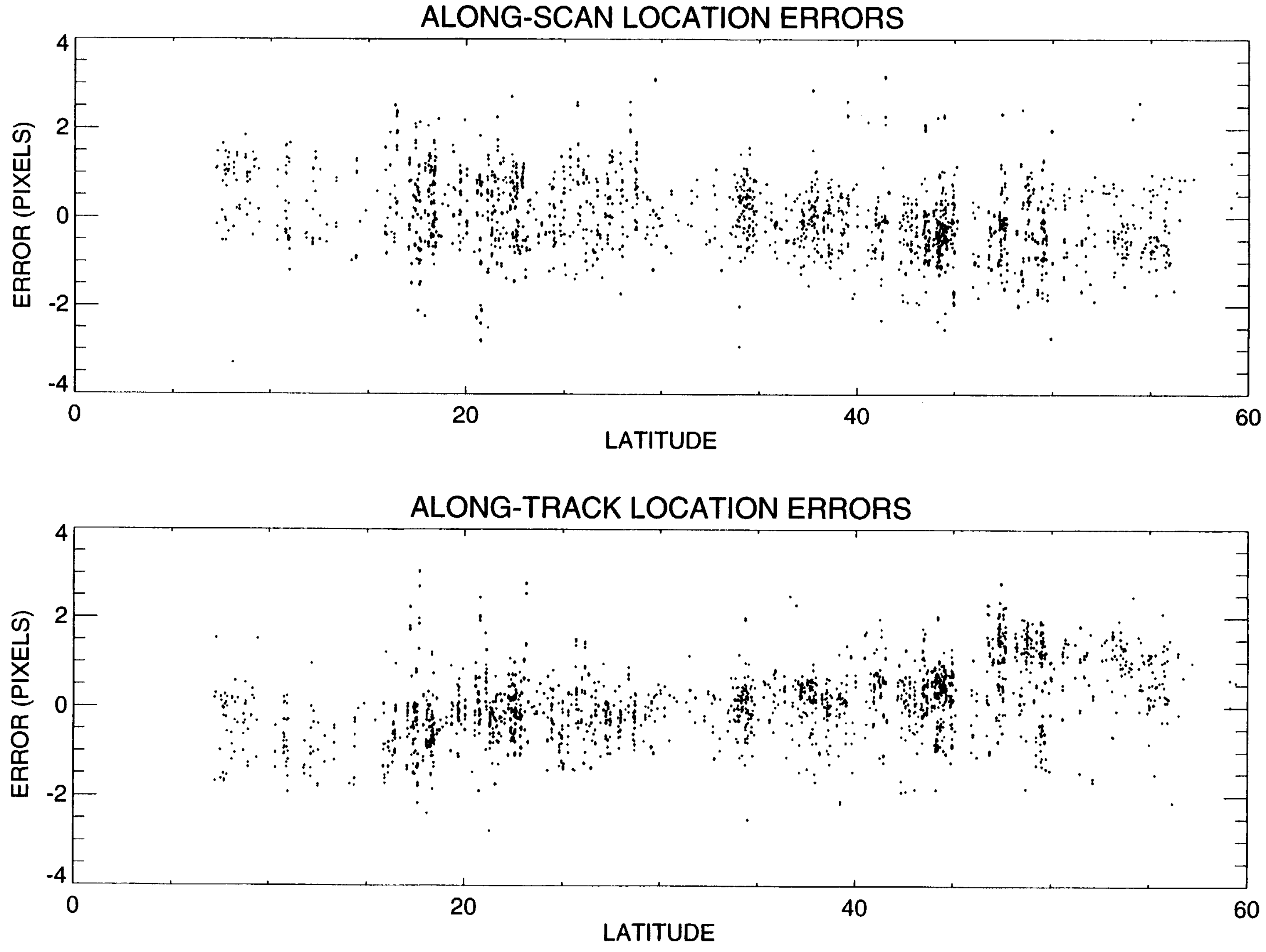


Figure 4 -- Island Location Errors vs. Latitude: Days 30 - 39 of 1998

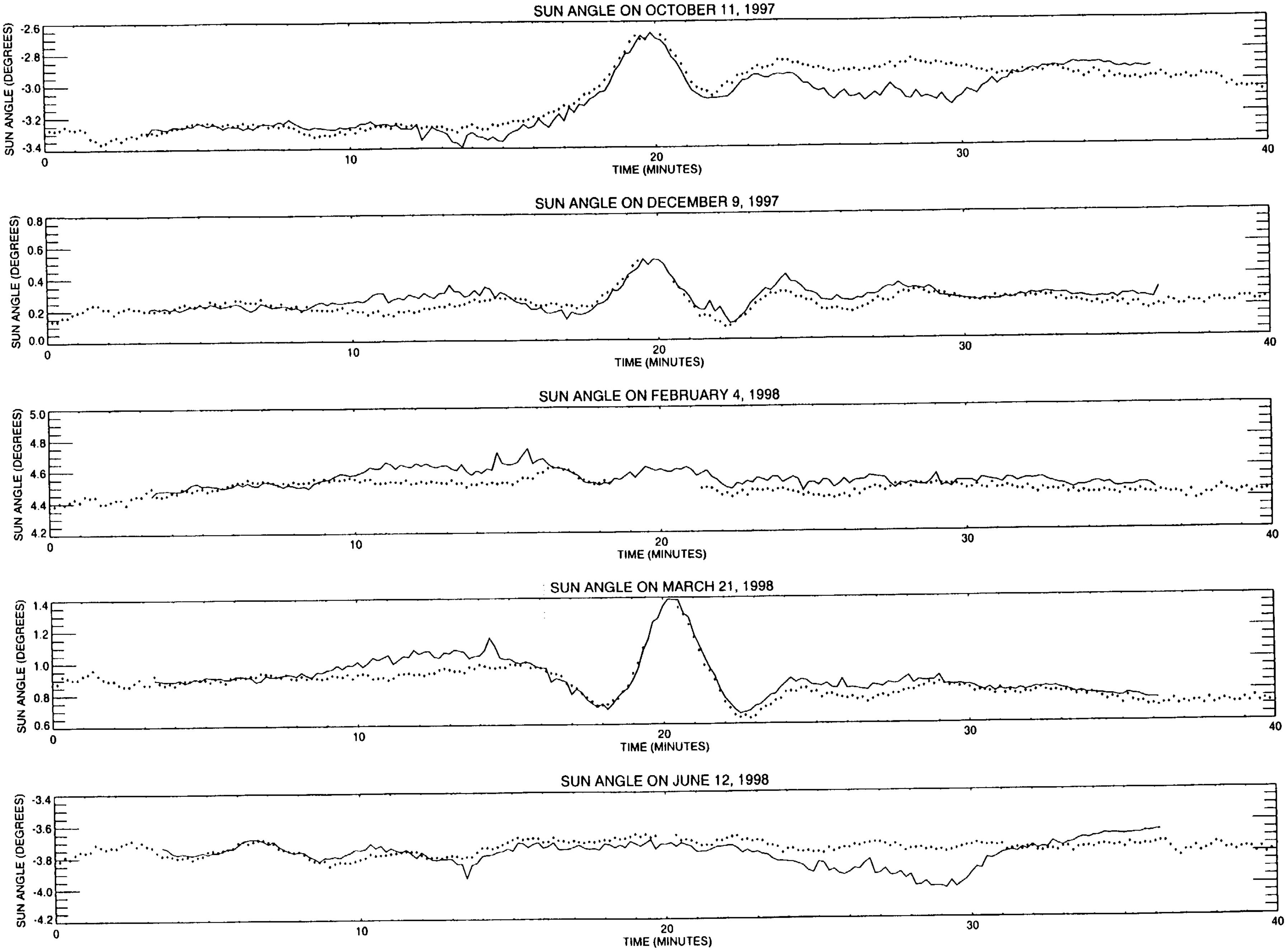


Figure 5 -- Sun Angle to Spacecraft X-Y Plane for DSS-A, DSS-B and DSS-C

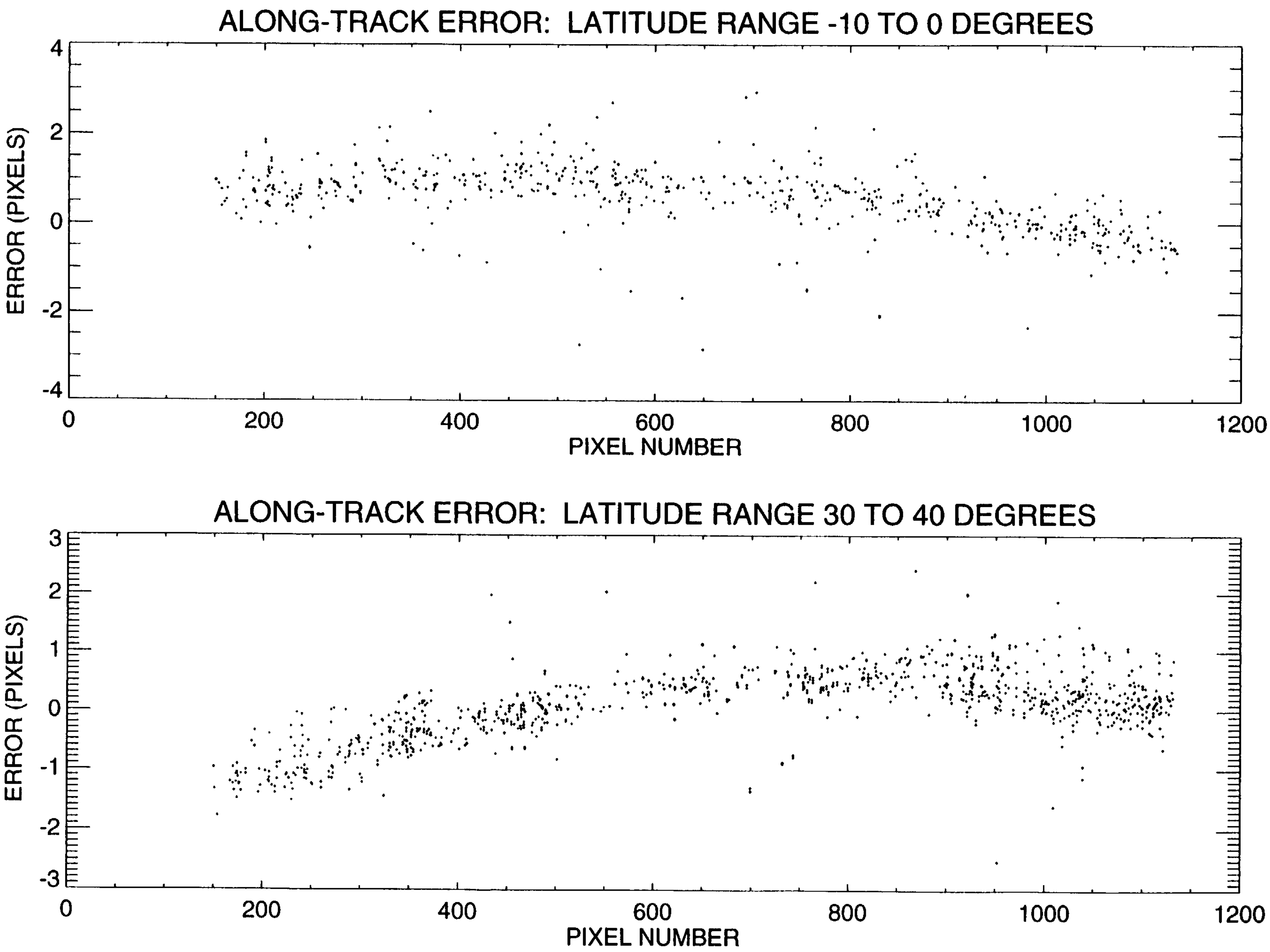


Figure 6 -- Island Location Errors vs. Pixel Number: Days 200 - 209 of 1998

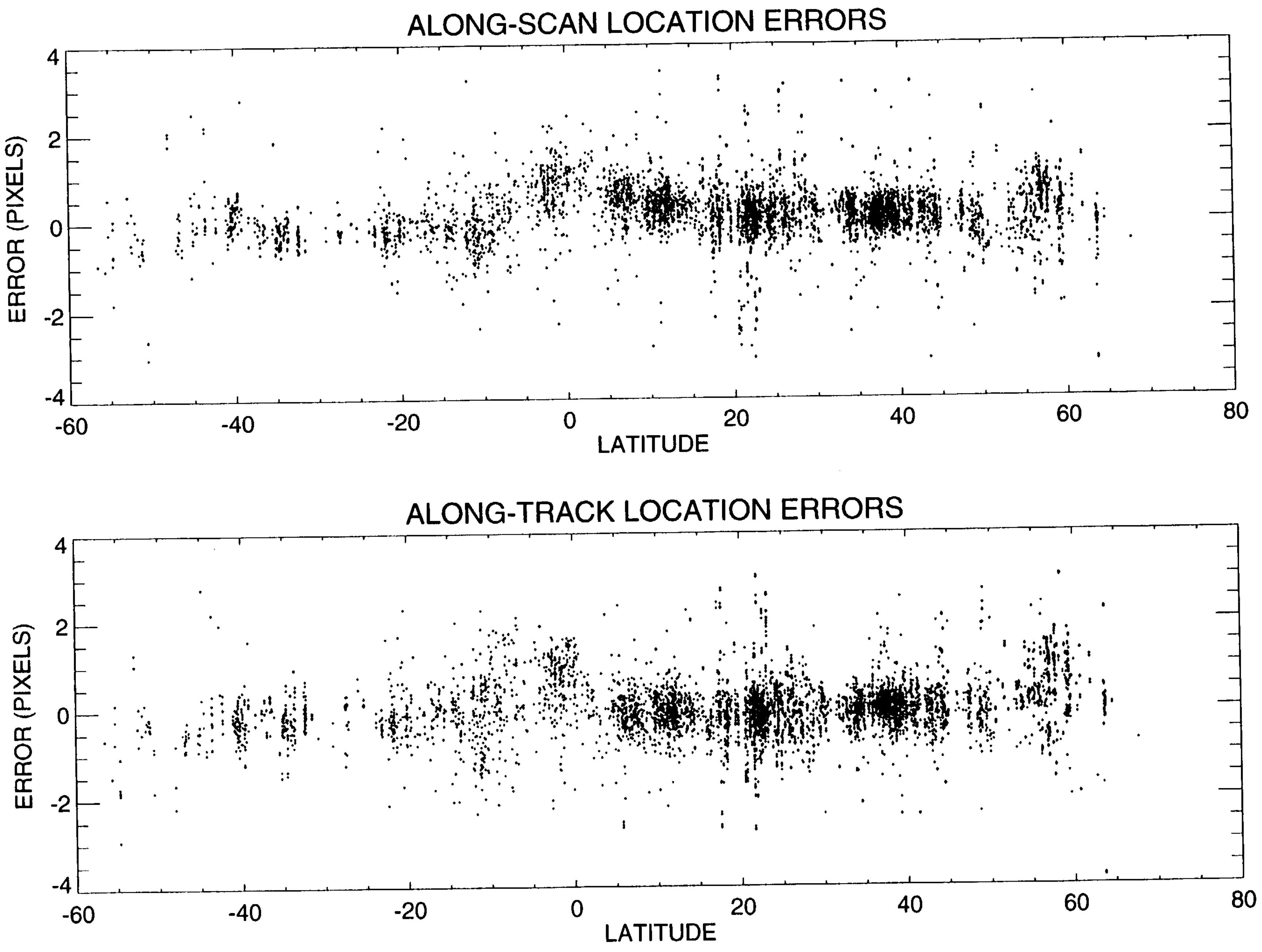


Figure 7 -- Island Location Errors vs. Latitude: Days 60 - 69 of 1999

Precise Orbit Determination for the GEOSAT Follow-On Spacecraft[1]

Frank G. Lemoine, David D. Rowlands
Space Geodesy Branch, Code 926, NASA GSFC

Nikita P. Zelensky, Scott B. Luthcke, Christopher M. Cox
Raytheon ITSS Corp., Greenbelt, MD

Gregory C. Marr
Flight Dynamics Analysis Branch, Code 572, NASA GSFC

ABSTRACT

The US Navy's GEOSAT Follow-On spacecraft was launched on February 10, 1998 with its primary mission objective to map the oceans using a radar altimeter. The spacecraft tracking complement consists of GPS receivers, a laser retroreflector and Doppler beacons. Since the GPS receivers have not yet returned reliable data, the only means of providing high-quality precise orbits has been though satellite laser ranging (SLR). SLR has tracked the spacecraft since April 22, 1998, and an average of 7 passes per day have been obtained from US and foreign stations. Since the predicted radial orbit error due to the gravity field is only two to three cm, the largest contributor to the high SLR residuals (10 cm) is the mismodelling of the non-conservative forces. The SLR residuals show a clear correlation with beta prime (solar elevation) angle, peaking in mid-August 1998 when the beta prime angle reached -80 to -90 degrees. We report in this paper on the analysis of the GFO tracking data (SLR, Doppler, and if available GPS) using GEODYN, and on the tuning of the non-conservative force model and the gravity model using these data.

INTRODUCTION

The launch of the GEOSAT Follow-On (GFO)[2] satellite February 10, 1998, marks the beginning of the Navy program to develop an operational series of low-cost altimeter satellites for maintaining continuous ocean observation via the GEOSAT exact repeat orbit (Table 1). GFO provides real-time measurements of the relative ocean heights for tactical applications and absolute heights post-processed for large-scale ocean modelling.

GFO carries a single frequency (13.5 GHz) radar altimeter, a dual frequency water vapor radiometer, a dual frequency Doppler beacon for operational tracking, a laser retroreflector and four Global Positioning System (GPS) dual-frequency receivers for precision orbit determination (POD).

The measured quantity of interest, the ocean surface above the reference ellipsoid, is in fact a combination of two measurements: the ocean surface with respect to the satellite as observed by the altimeter, and the satellite height above the reference ellipsoid determined from the satellite tracking. GFO's capability to produce precise observations of the ocean surface thus depends critically on the accuracy of the orbits produced from the Doppler, SLR, or GPS tracking (Table 2).

Since the GPS receivers have yet to deliver reliable data[1], SLR tracking has provided the only means for computing highly accurate orbits. The 5-cm radial orbit error estimate for SLR tracking shown in Table 2 was derived in a pre-launch simulation study [2]. It is the Root Mean Square (RMS) error over one day. Orbits with somewhat better accuracy (2.2 cm) could be computed using GPS data. The study (Ref. 2) also indicates that error in GFO precise orbits is driven by force model error.

The Space Geodesy Branch at Goddard Space Flight Center (GSFC) has been given the task of improving GFO POD. This work has included pre-flight orbit error analysis and now involves tuning a "macro model" of the approximate spacecraft geometry and surface properties in order to better model the nonconservative forces, as well as tuning the gravity model. The GEOSAT exact repeat orbit is designed to observe the same groundtrack every 17 nodal days. Orbit maneuvers are carried out periodically to

[1] The authors acknowledge Yoaz Bar-Sever of the Jet Propulsion Laboratory and Scott Mitchell of Ball Aerospace for information pertaining to modelling of the GFO attitude. J. Andrew Marshall (now at Lockheed-Martin, Denver) developed the preliminary macro-model for GFO.

[2] Abbreviations and acronyms used in this paper are compiled in Appendix A.

maintain the same groundtrack to within 1 km.

This paper reviews the analysis of GFO POD using the untuned models, and presents evidence that the largest source of orbit error is due to mismodelling of the solar radiation pressure. It is believed that tuning the macro model will correct this and a preliminary analysis of such tuning is presented.

GFO PRECISION ORBIT DETERMINATION (POD) AND ANTICIPATED ERRORS

Orbit accuracy depends on POD model accuracy, quality of the tracking data, and choice of solution strategy. GEODYN[3], a state-of-the-art least squares orbit determination and geodetic parameter recovery GSFC program, is used for GFO POD. Table 3 offers a summary of the POD models. Several gravity fields were tested, EGM96[4], TEG3[5], JGM3[6], and PGS7609G, a recent GSFC combination model based on EGM96 but with additional TDRSS satellite tracking data from the EUVE, ERBS, XTE, GRO, and TRMM satellites.

Although covariance projections indicate that orbit error due to gravity will be only 3-5 cm (Table 4), the error structure will be complex, and include a geographically correlated component. By spherical harmonic order, the radial orbit error due to gravity is highest at order 1, and in the vicinity of the k=2 resonance (near order 29) (Figure 1). Tuning with GFO tracking data should reduce this error.

Nonconservative forces acting on GFO consist of radiative forces and atmospheric drag. Radiative forces include solar radiation pressure, the Earth's albedo (reflected light) and infrared radiation, and other secondary effects such as thermal imbalance in emission from spacecraft surfaces. The macro model approximates GFO's surface geometry and material properties using eight plates (Figure 2). Each plate has been assigned a body-fixed orientation, area, and specular and diffuse reflectivity coefficients based on pre-launch engineering information. All plate interaction effects, such as shadowing and multiple reflections, are ignored. The total acceleration with respect to the center of mass (CoM) is computed by summing vectorially the force acting on each plate, taking into account each plate's area, angle of incidence and material properties. Throughout the orbit and over a Beta prime cycle, radiation will be incident to a changing orientation of the macro model as computed using an analytical attitude model. Beta prime is the angle to the sun from the orbit plane (Figure 3), and for GFO shows a period of about 336 days (Figure 4). The spacecraft mass is assumed to be 369 kg.

As shown in Figure 5, the largest nonconservative force acting on GFO is by far due to solar radiation pressure. Since the solar radiation pressure is so large, even a small error will have a significant impact. The error for the untuned macro model will be 10 to 20 percent of the radiative force. For instance the *a priori* macro model for TOPEX was meticulously constructed using finite element modelling and could only account for 90% of the radiative forces[7]. However, after tuning, the TOPEX macro model is believed to account for over 95% of the radiative forces[8].

The measurement model must first be verified before proceeding to refine the force models. Two vectors are involved for defining the Laser Retroreflector Array (LRA) position with respect to the spacecraft CoM: 1) the location of the LRA phase center with respect to the spacecraft body-fixed coordinate system, and 2) the location of the Center of mass in this coordinate system. The LRA is fixed and only the location of the CoM changes with time, based on propellant usage. We estimated the LRA phase center from SLR tracking, and this estimate would accommodate to first order changes in the CoM for which we do not have detailed information. The LRA consists of nine corner cubes arranged hemispherically[9], and is expected to have a stationary phase center, independent of the tracking geometry. An analysis of measurement biases computed from one month of SLR residuals, does not show any correlation with elevation angle.

Errors in the attitude model will not only affect the computation of the nonconservative forces, but more directly the modelled observation. The LRA orientation changes according to the attitude profile. As in all radar altimeter satellites, GFO is nadir pointing, and with attitude modelling defined only in the yaw. The analytical GFO yaw attitude model follows a cosine function with a period of one orbit revolution and whose amplitude is determined by the Beta prime angle (Figure 6). On several occasions, the implemented analytical model has been checked against especially requested telemetered attitude angles and has shown little discrepancy. For example, over August 8 1998, there is a Root Mean Square (RMS) discrepancy of 0.206°, 0.139°, and 0.248° for the roll, pitch, and yaw axes respectively (Figure 7). This can map to an SLR RMS measurement error of no larger than 0.5 cm, which is near the expected 0.3-.05 cm SLR measurement noise level of third generation systems.

TRACKING DATA AND ORBIT ANALYSIS

GFO POD relies on SLR tracking provided by a global network of NASA and foreign stations (Figure 8). Unfortunately the tracking has been sparser than anticipated showing an average of about seven passes per day (Figure 9). Operational tracking Doppler data from the three stations (Guam, Point Mugu California, and Prospect Harbor Maine) although noisy is also abundant, and serves to slightly strengthen the SLR solution. After 40% of the data is edited, we typically use nine Doppler passes per day. The Doppler station positions have been adjusted to the SLR frame using three months of Doppler data and SLR-determined orbits that were held fixed in the solution.

Given the tracking density, an arc length of five days was selected over shorter arcs to increase the dynamic strength of the solution[3]. Arc lengths of nine and ten days would also be suitable, however the frequency of satellite events such as computer resets or maneuvers which are not modelled in GEODYN for POD, have allowed only a few uninterrupted ten day spans.

The solution strategy, with the objective to minimize orbit error, was developed considering the strength of the tracking data. Several parameterization schemes were tested and the one finally selected is summarized in Table 3. According to this strategy orbit error is minimized by adjusting, in addition to the orbit state, one coefficient (lightly constrained) to scale the solar radiation pressure, a daily atmospheric drag scale coefficient, and an empirical one cycle per revolution (1cpr) acceleration for both the along-track and cross-track components. The empirical terms absorb much of the residual accelerations remaining from the mismodelling of the various forces and greatly reduce orbit error[10]. Since the empirical acceleration terms capture information about the residual accelerations they should be left out of solutions designed to tune say the macro model, but in this case may be used to reveal the characteristics of the mismodelled forces.

The initial orbit solutions show a large SLR residual mean, suggesting a measurement modelling error. One month (June 1998) of data was sufficient to estimate the LRA offset (Table 5). It is not known which portion of the adjustment is due to an error in the LRA offset or to an error in the specification of the CoM in the spacecraft coordinate system, but application of the estimated offset reduces the SLR fits and the residual mean over several months tested (Tab.5, Figure 10).

No single test can uniquely gauge orbit accuracy. This analysis uses SLR residuals, or the misclosure between the highly precise observed and computed ranges, and orbit differences between arcs sharing one day of overlapping data, to indicate the level of orbit error. Since orbit error is typically minimized in the middle of an arc, overlap orbit differences computed over the ends of the arcs should represent a reasonable estimate of error. GFO altimeter crossover residuals may also be used to test for orbit error.

We see considerable variation in the SLR fits from arc to arc with an overall fit of about 10-cm (Figure 11). Gravity error is unlikely to explain the variation in orbit fits, as the four gravity fields tested show similar fits per arc, and more importantly manifest the same trend over time. The change in the GFO orbit fits (for all gravity fields tested) correlates with an increase in the absolute value beta prime angle. The orbit fits peak at the largest absolute beta prime values. This trend in SLR orbit residuals is consistent with a mismodelling of the radiative forces acting on the GFO spacecraft. The radial orbit differences between adjacent overlapping arcs shows a similar degradation at high (in absolute value) beta prime angles (Figure 12 and Table 6).

As Beta prime decreases from zero to -84 degrees the solar radiation pressure (and the mismodelled effect) will change its projection from the predominately along-track and radial directions to the cross-track (Figure 13). The adjusted empirical accelerations should thus decrease in magnitude in the along-track component, and increase in the cross-track. This is just what we see (Figure 14). Not only does the along-track 1cpr empirical acceleration magnitude decrease, but the associated phase (with respect to orbit angle) remains constant from arc to arc until the spacecraft enters the full sunlight regime (Figure 15). The observed phase coherence indicates that the force error preserves the same orientation with respect to orbit plane from arc to arc, which in fact solar radiation pressure does (Figure 13). As the spacecraft reaches full sunlight (near -65° beta prime), the recovered along-track acceleration magnitude becomes very small, for which the phase is probably not well determined. In another study tuning the TDRSS macro model[11], a continuous phase was also observed in the recovered 1cpr along-track acceleration prior to tuning. After tuning, the recovered acceleration magnitudes were small and the phases showed no coherence.

Since nonconservative force model error dominates, tuning the macro model will precede tuning of the gravity field. Use of a macro model tuned over the beta prime cycle of about 336 days should minimize nonconservative force model error

[3] In cases of sparse tracking one can usually rely on the fidelity of the dynamic force models to determine a better orbit over a longer span. Over a shorter arc, the solution may be ill determined, and the orbit error very large over periods with no data.

contamination in gravity field tuning.

MACRO MODEL TUNING

The macro model represents the GFO spacecraft as an eight surface composite (Fig 2). It approximates the spacecraft geometry and surface material properties to better model the surface force effects due to solar and terrestrial radiation pressure, and due to atmospheric drag. Each surface (or plate) had been assigned an orientation with respect to the satellite fixed frame, an area, and a specular and diffuse reflectivity coefficient based on pre-launch engineering specifications. The material properties of each plate are assumed to be homogenous, representing an average value. In tuning, these average values are adjusted to best fit the GFO tracking data using an orbit determination (OD) solution strategy to insure the mismodelled nonconservative forces are not absorbed in empirical parameter adjustments. Therefore the macro model is tuned to the residual satellite acceleration history which is based on orbit errors sensed from the spacecraft tracking data.

OD parameterization suitable for macro model tuning adjusts the orbit state, the solar radiation pressure coefficient (C_R), and one drag coefficient (C_D). Upon solution convergence for each arc, GEODYN writes out the normal equations for the orbit (state, C_R, C_D) and panel (area, specular, diffuse) parameters. These normal equations will be combined from arcs sampled over the beta prime cycle and the selected panel parameters estimated using Bayesian least squares. The tuning procedure at GSFC, refined with the experience gained tuning the TOPEX (Ref 9) and TDRSS (Ref 11) macro models, has been previously described in Ref 11.

A preliminary sensitivity study was performed using the combined normal matrix from four well-spaced arcs to help identify panel parameters that are to be estimated. Assuming a specified allowed percent change in each respective panel parameter *a priori* value, and using only the left-hand side diagonal (variance) terms of the normal matrix, the resulting "uncorrelated weighted variance" is computed in order to compare parameter sensitivity or change in residual variance with respect to parameter adjustments. The *a priori* surface area assigned to each plate is believed to be relatively well determined with about a 10% error. There is much greater uncertainty for the *a priori* specular and diffuse reflectivity coefficients, computed as an aggregate average of these properties for each surface. The area is allowed to change by 10% and the reflectivity coefficients by 100% for the sensitivity analysis. As shown in Figure 16, specular coefficients for four parameters representing the solar array, the bottom plate (+z facing Earth), and the top and bottom sides of the altimeter antenna reflector, are likely candidates for the macro model tuning adjustment.

From previous experience, it is anticipated tuning will greatly reduce the radiative force model error. Following tuning, atmospheric drag may become the dominant nonconservative force model error. Atmospheric drag, modelled as a function of atmosphere density and area projected in the along-track direction, is significant for GFO POD, in spite of the small relative magnitude (Fig. 5), due to the large uncertainty in the computed air density. The orbit error due to drag may be somewhat reduced by also adjusting the panel area parameters during tuning.

SUMMARY

The Space Geodesy Branch at Goddard Space Flight Center (GSFC) has been given the task of improving GFO precision orbit determination (POD). Pre-flight orbit error analysis preceded the current work which involves tuning a "macro model" of the approximate spacecraft geometry and surface properties in order to better represent the nonconservative forces, as well as tuning the gravity field. Since GPS receivers have yet to deliver reliable tracking data, SLR tracking offers the only means with which to compute precise orbits for this spacecraft. The current state-of-the-art POD models, as implemented in GEODYN, were employed to compute orbits using SLR tracking. The orbits, tracking data, and current POD models were evaluated and the SLR measurement modelling was verified. The attitude model was compared with telemetered values. SLR tracking data was used to estimate the LRA phase center to CoM vector.

Analysis of the SLR orbit fits, orbit differences between adjacent overlapping arcs, and recovered empirical accelerations indicate a radial orbit error of about 10-cm dominated by solar radiation pressure. For this reason tuning of the macro model will precede tuning of the gravity field. A preliminary sensitivity analysis of the macro model parameters suggests that the specular coefficients for the solar array, the spacecraft body bottom plate, and altimeter antenna reflector are the best candidates for the tuning adjustment. Based on experience in tuning models for TOPEX and TDRSS, it is anticipated that tuning the macro model, and subsequently the gravity field using GFO tracking data will significantly reduce the current levels of orbit error for GFO.

APPENDIX A

ABBREVIATIONS AND ACRONYMS

Abbreviations found in this paper are normally defined after the first use of the technical term.

CoM	center of mass
ERBS	Earth Radiation Budget Satellite
EUVE	Extreme Ultra-Violet Explorer
GEOSAT	Geodesy Satellite
GFO	GEOSAT Follow-On
GPS	Global Positioning System
GRO	Compton Gamma Ray Observatory
GSFC	Goddard Space Flight Center
LRA	laser retroreflector array
NASA	National Aeronautics and Space Administration
OD	orbit determination
POD	precision orbit determination
RMS	root mean square
SA	solar array
SLR	satellite laser ranging
TDRSS	Tracking and Data Relay Satellites System
TOPEX	Ocean Topography Experiment
TRMM	Tropical Rainfall Measuring Mission
XTE	Bruno B. Rossi X-ray Timing Explorer

TABLES AND FIGURES

Table 1. GEOSAT Exact Repeat Orbit

Orbit parameter	value
Altitude	800 km
Eccentricity	0.0008
Inclination	108 deg
Repeat Period	244 revs in 17 days

Table 2. GFO Altimeter Error Budget[3]

Component	Source	Error (cm)
Altimeter		
instrument noise	Ball	1.9
biases	Ball	3.0
sea surface (EM & skewness)	TOPEX	2.3
Media		
troposphere	Ball	2.0
ionosphere	Ball	1.7
SLR POD (radial orbit)	GSFC	5.0
Total RSS	▨	7.1

Table 3. GFO Precise Orbit Determination Modelling

Model Category	Description
Geophysical models	
Gravity	PGS7609G (EMG96+ TDRSS tracking of GRO, XTE, TRMM, ERBS)
Ocean/Earth Tides	PGS7609G (based on Ray '94 [12])
Atmospheric density	MSIS-86[13]
Spacecraft geometry and surface forces	GFO *a priori* macro model
Station Coordinates	CSR95L02 SLR solution frame[5]
Earth Orientation Parameters	CSR95L02 from LAGEOS tracking
Planetary Ephemeris	DE403
Measurement Model	
SLR	*A priori* CoM, estimated LRA offset, analytical attitude
Doppler	*A priori* CoM, *a priori* beacon offset, analytical attitude
Tracking Data Weights	
SLR	10 cm
Doppler	2 cm/sec
Estimated Parameters	Orbit state, Atmospheric drag C_D per day Solar radiation pressure C_R per arc (lightly constrained) Along-track 1cpr empirical acceleration per arc Cross-track 1cpr empirical acceleration per arc Doppler measurement and troposphere bias per pass

[4] An official GFO altimeter system error budget has yet to be published. The values shown here have been compiled from an internal Ball document provided by Scott Mitchell, from the TOPEX/POSEIDON Mission Plan, and the orbit error simulations performed at GSFC.

[5] CSR95L02 is the SLR station position and velocities used to compute the TOPEX/POSEIDON precise orbits, Richard Eanes, CSR, 1995.

Table 4. Gravity Orbit Error Covariance (to 70x70) Projection for GFO

Gravity Field	**GFO Orbit Error (cm)** Radial	Cross-Track	Along-Track
JGM3	4.97	23.80	42.61
TEG3	3.48	21.42	42.76
EGM96	2.61	8.94	17.72
PGS7609G	2.61	8.93	16.44

Table 5. LRA Offset - CoM Vector Estimate

Description	**Spacecraft Body-fixed Coordinates (cm)** X	Y	Z	**SLR residual Mean (cm)**	**SLR residual RMS (cm)**
A priori CoM	89.7	0.8	-6.6		
A priori offset	114.2	77.2	42.7	-2.5	10.7
Estimated offset	107.9	76.1	53.3	-0.1	10.0

Table 6. SLR Fit and Orbit Precision in relation to Beta Prime

Beta prime regime Absolute value (degrees)	**SLR fit (cm)**	**Orbit Overlap Arc Difference (cm)**		
		Radial	Cross-Track	Along-Track
0 to 64 (low)	9.4	7.7	20.7	56.6
65 to 85 (high)	10.4	14.6	23.5	76.2
Total	10.0	10.8	21.8	64.7

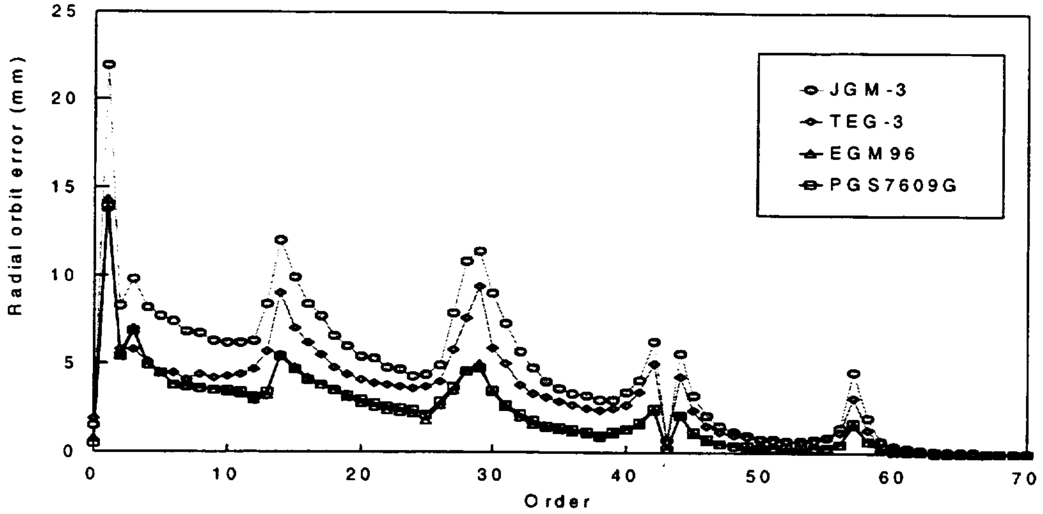


Figure 1. GEOSAT radial orbit error from gravity field covariances to 70x70

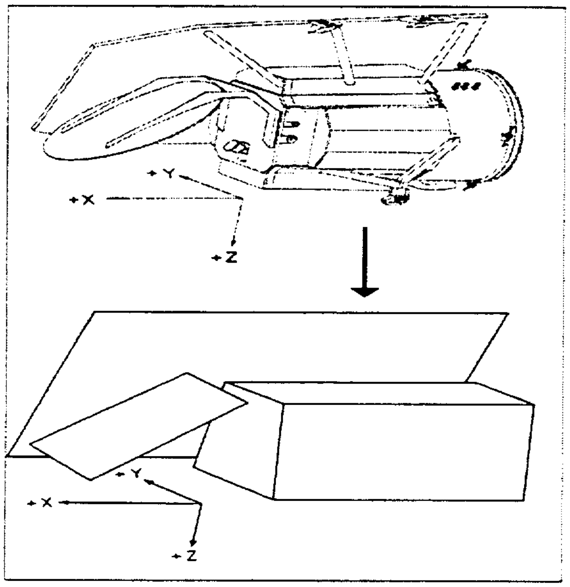


Figure 2. GFO spacecraft macro model approximation

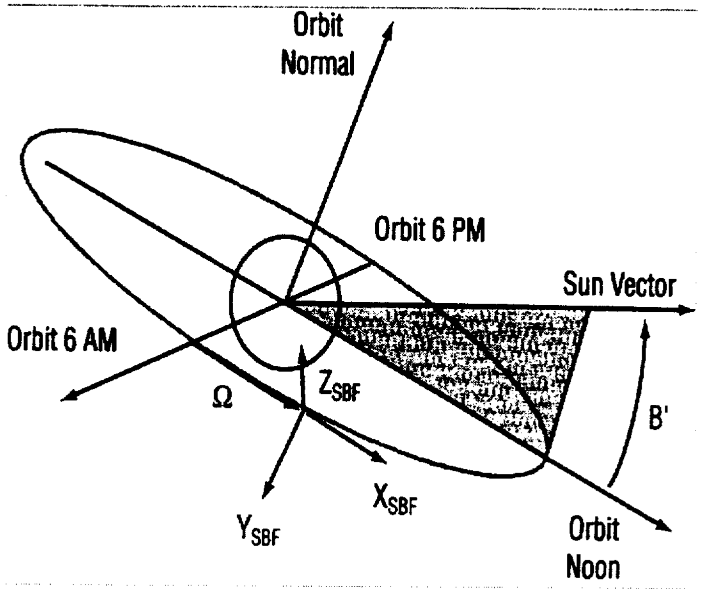


Figure 3. Orbit geometry

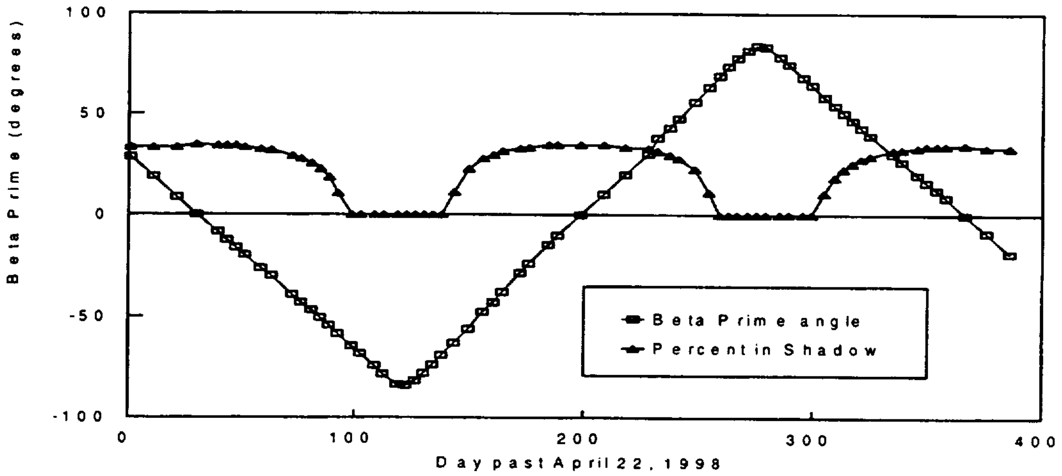


Figure 4. GFO Beta Prime Cycle

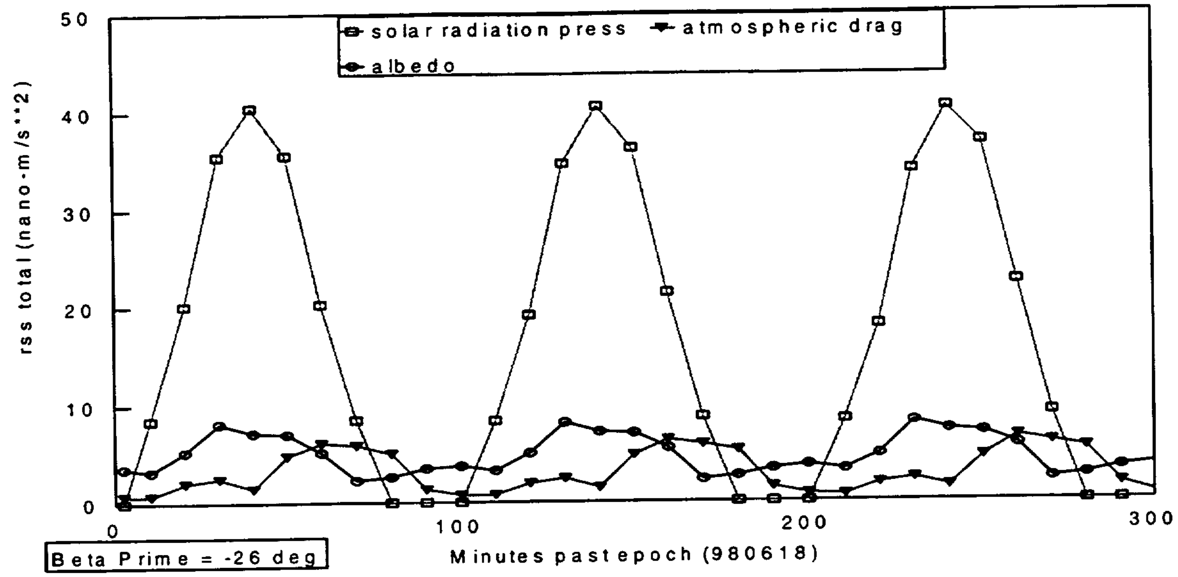


Figure 5. GFO surface force accelerations

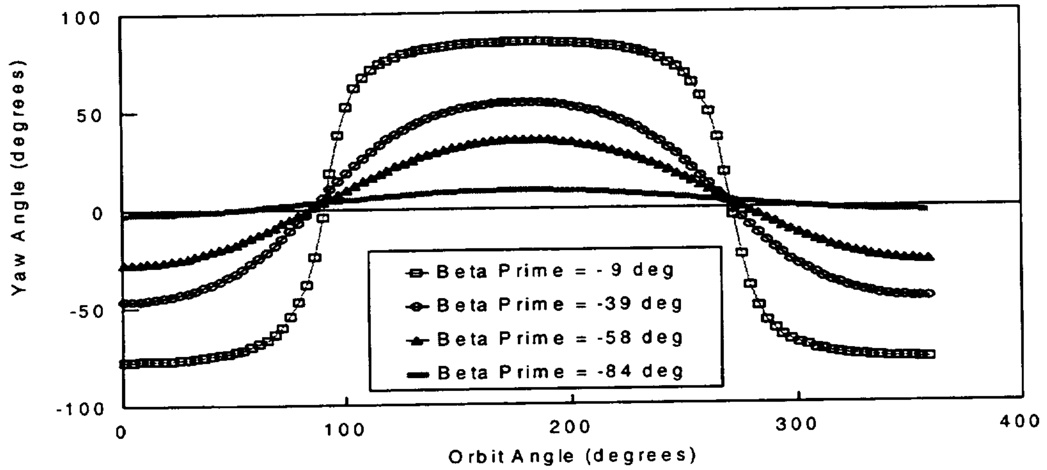


Figure 6. GFO yaw attitude model

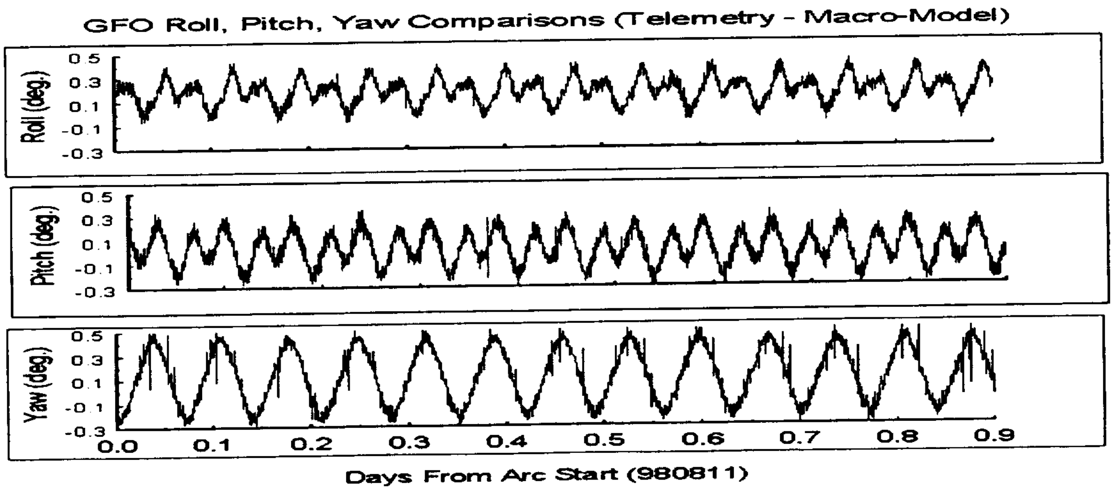


Figure 7. GFO roll, pitch, yaw comparisons (telemetry-macro model)

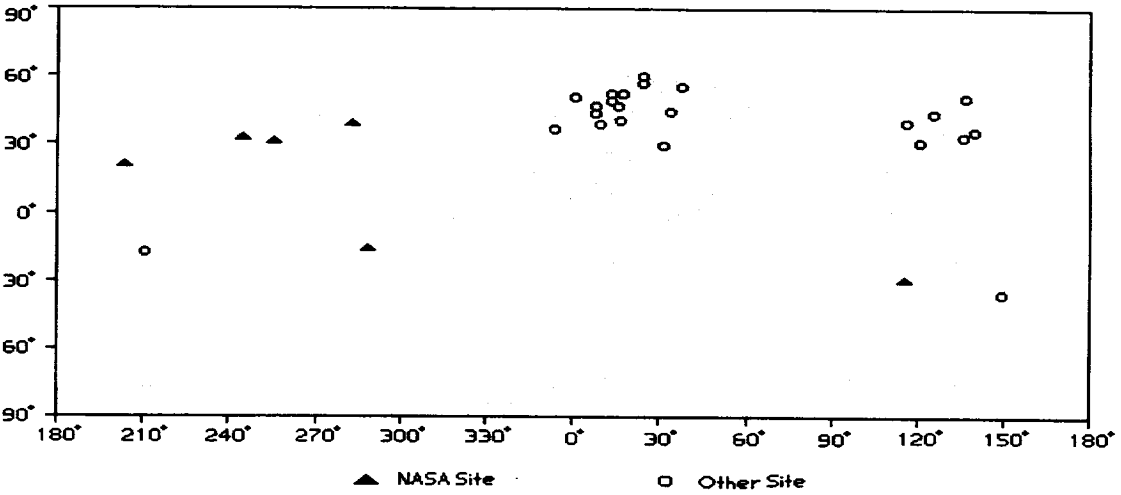


Figure 8. GFO SLR tracking network

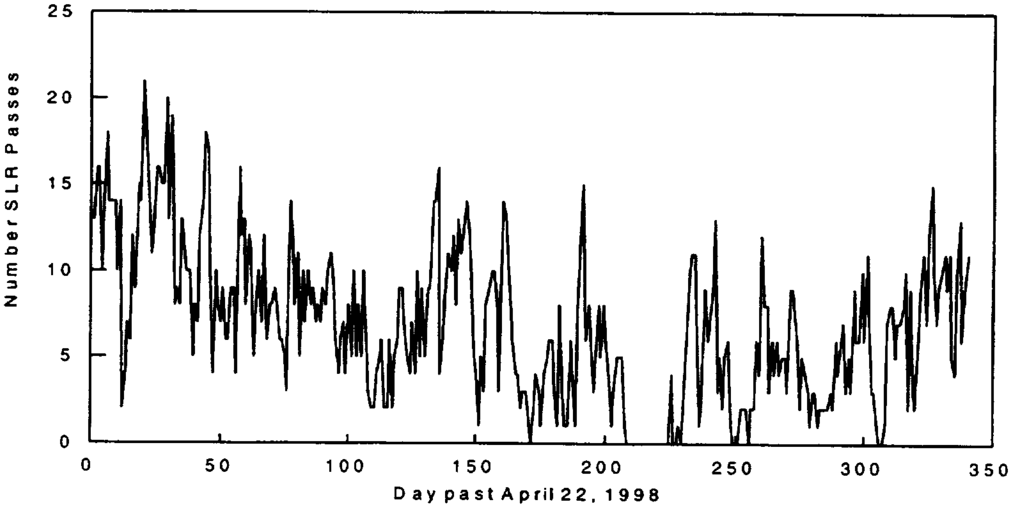


Figure 9. GFO SLR tracking history

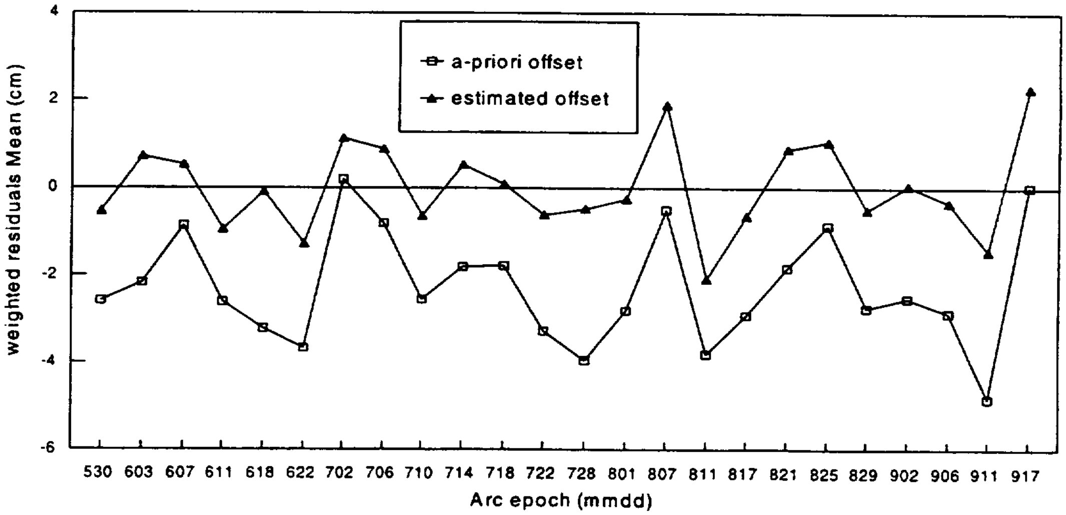


Figure 10. GFO SLR residual means and LRA offset

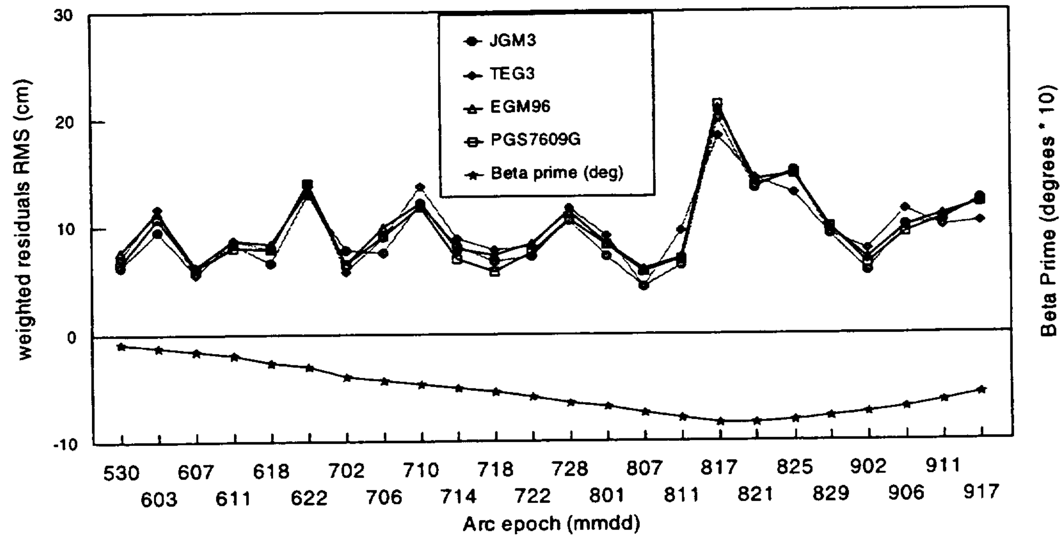


Figure 11. GFO SLR fit

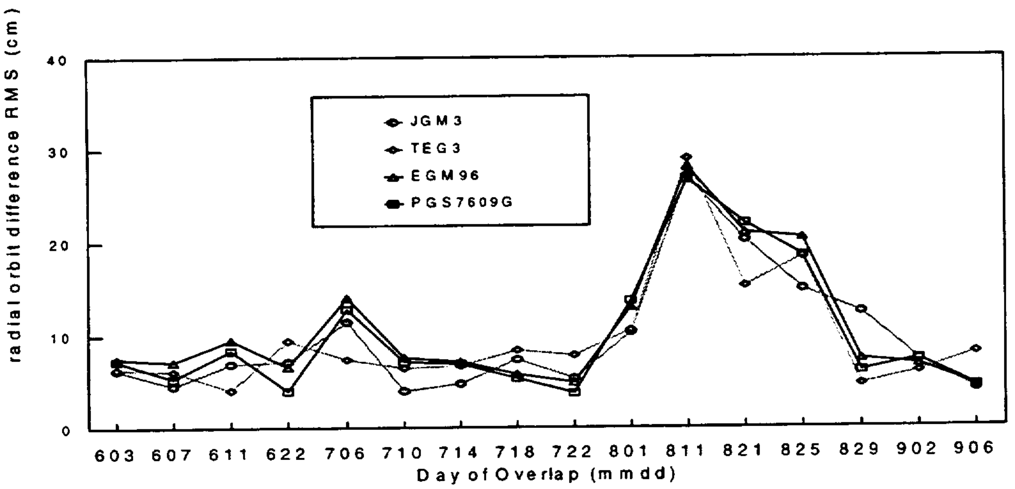


Figure 12. GFO radial orbit difference for overlapping arcs

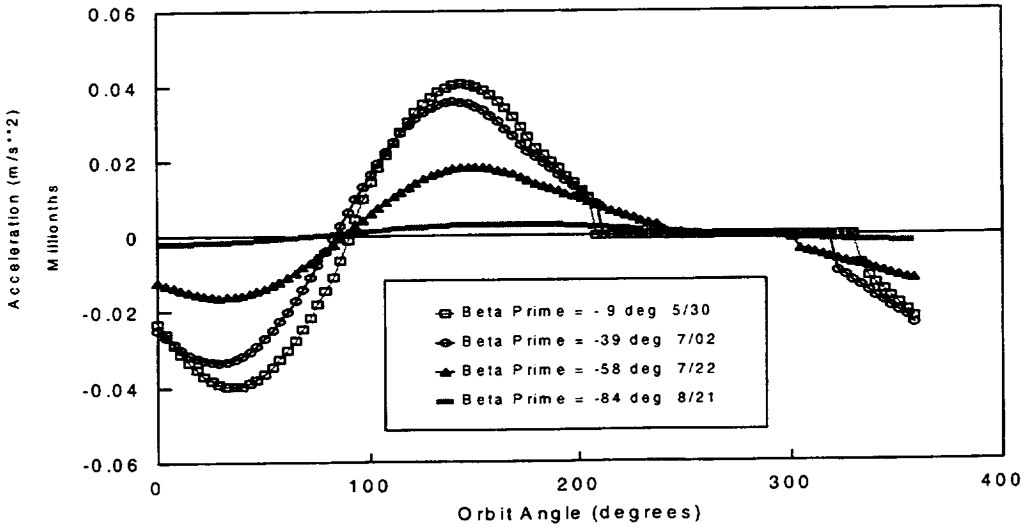


Figure 13. GFO solar radiation pressure along-track

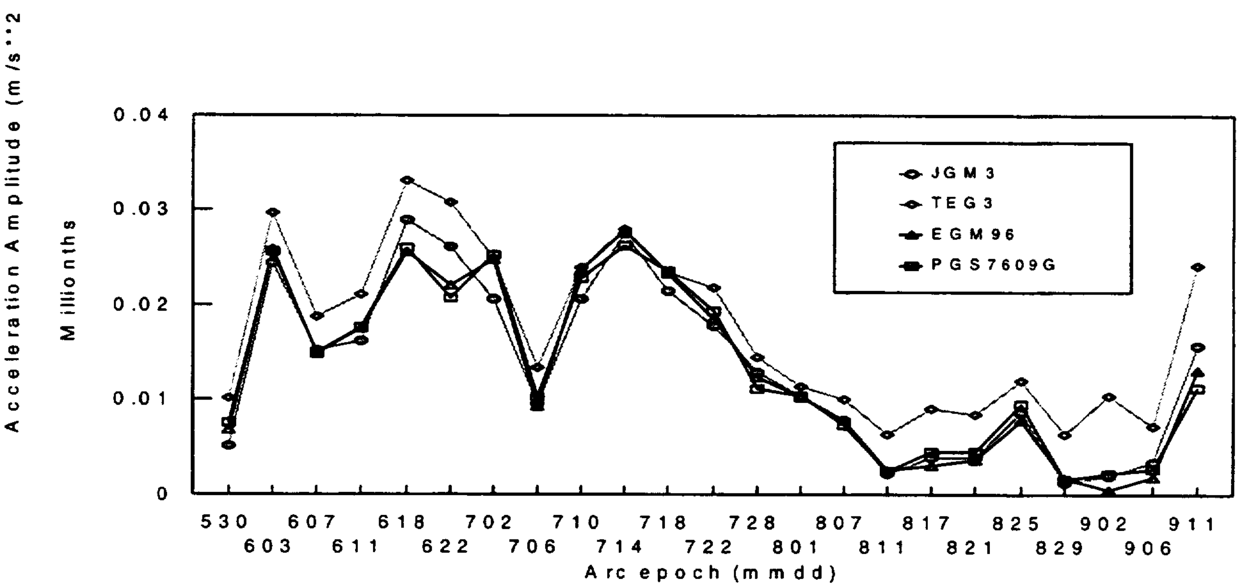


Figure 14. GFO estimated 1cpr along-track amplitude

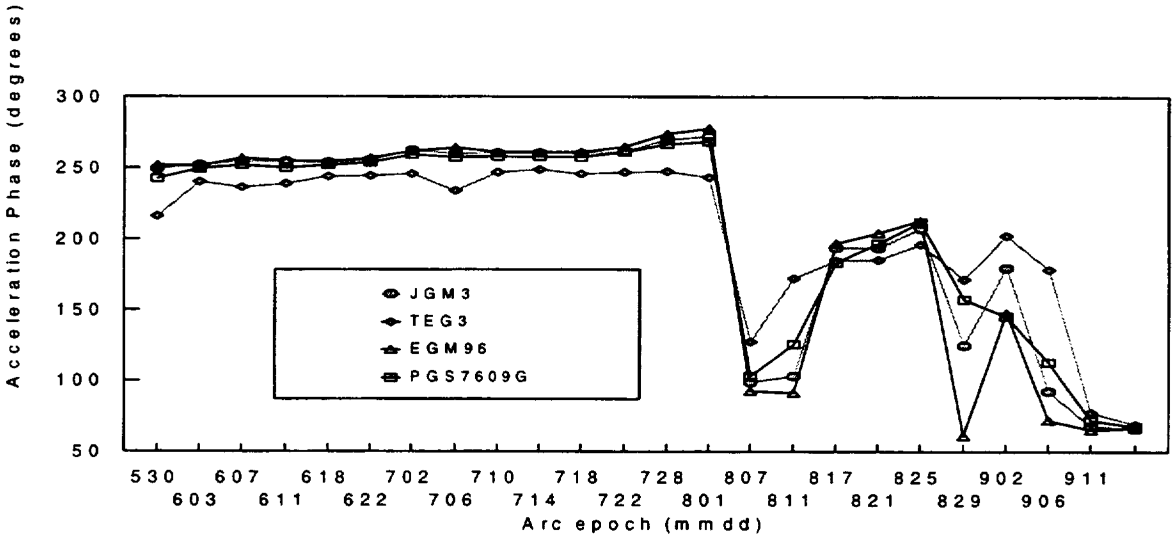


Figure 15. GFO estimated 1cpr along-track phase

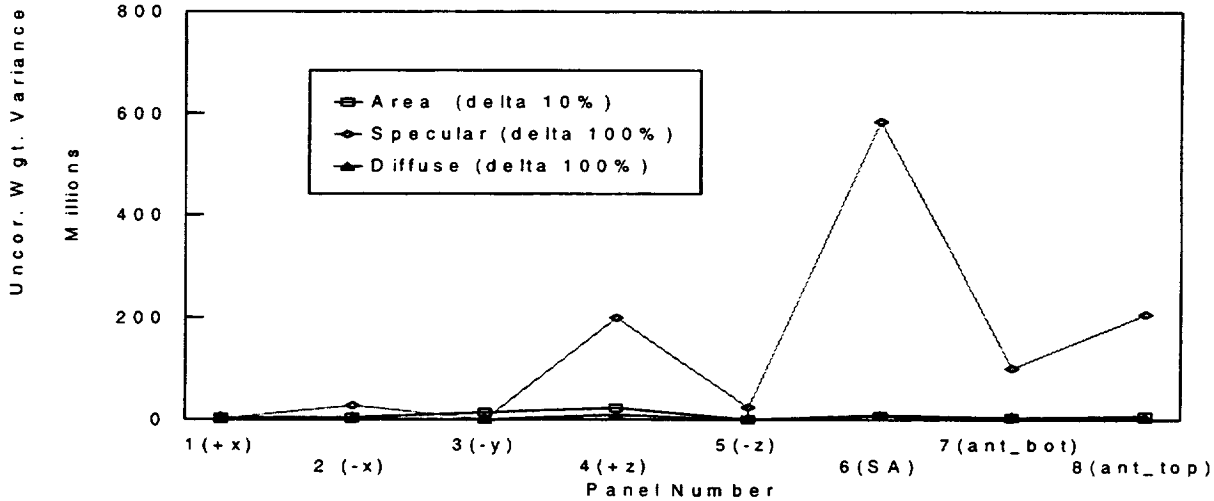


Figure 16. GFO macro model parameter sensitivity (uncorrelated weighted variance)

REFERENCES

1. Frazier W., Mitchell S., Weiss M., Wiener D., "Initialization and Early On-Orbit Performance of the Geosat Follow-On Satellite", *22nd AAS Guidance and Control Conference*, Feb '99, Breckenridge, Colorado, AAS 99-072.

2. Zelensky N.P., Luthcke S.B., Gehrmann L., Rowlands D.D., Marshall J.A., Lemoine F.G., "Error Analysis of the GEOSAT Follow-On Satellite Orbit Determined Using SLR and GPS Tracking (abstract)," *Annales Geophysicae EGS XXII General Assembly*, Vol 15, suppl. 1, pp. C187, April 1997.

3. D.E. Pavlis. S. Lou, P. Dahiroc, J.J. McCarthy, and S.B. Luthcke, GEODYN operation manuals, five volumes, Raytheon ITSS, December 18, 1998.

4. F.G. Lemoine, S.C. Kenyon, J.K. Factor, R.G. Trimmer, N.K. Pavlis, D.S. Chinn, C.M. Cox, S.M. Klosko, M.H. Torrence, Y.M. Wang, R.G. Williamson, E.C. Pavlis, R.H. Rapp, and T.R. Olson, *The Development of the Joint NASA GSFC and NIMA Geopotential Model EGM96*, NASA/TP-1998-206861, July, 1998, NASA Goddard space Flight Center, Greenbelt, MD.

5. B.D. Tapley, C.K. Shum, J.C. Ries, S.R. Poole, P.A.M. Abusali, S.V. Bettadpur, R.J. Eanes, M.C. Kim, H.J. Rim, and B.E. Schutz, "The TEG-3 geopotential model," in *Gravity, Geoid, and Marine Geodesy* J. Segawa, H. Fujimoto, and S. Okubo (eds.), Vol 117, *International Association of Geodesy Symposia*, Springer-Verlag, Berlin, 453-460, 1997.

6. B.D. Tapley, M.M. Watkins, J.C. Ries, G.W. Davis, R.J. Eanes, R. Poole, H.J. Rim, B.E. Schutz, C.K. Shum, R.S. Nerem, F.J. Lerch, J.A. Marshall, S.M. Klosko, N.K. Pavlis, and R.G. Williamson, The Joint Gravity Model-3, *J. Geophys Res.*, 101(B12), 28029-28049, 1996.

7. J.A. Marshall and S.B. Luthcke "Modeling Radiation Forces Acting on TOPEX/POSEIDON for Precision Orbit Determination," *Journal of Spacecraft and Rockets*, Vol 31, No. 1, 1994, pp. 99-105.

8. J.A. Marshall and S.B. Luthcke "Radiative Force Model Performance for TOPEX/POSEIDON for Precision Orbit Determination," *Journal of Astronomical Sciences*, Vol. 42, No. 2, 1994, pp. 229-246.

9. NASA SLR Satellite Information web page, http://ilrs.gsfc.nasa.gov/slr_sats/slr_satellites.html

10. Colombo, O.L., "Ephemeris errors of GPS satellites," *Bull. Geod.*, 60, 64-84, 1986.

11. S.B. Luthcke, J.A. Marshall, S.C. Rowton, K.E. Rachlin, C.M. Cox, R.G. Williamson, "Enhanced Radiative Force Modelling of the Tracking and Data Relay satellites", *JAS* Vol. 45, No. 3, July-September 1997, pp. 349-370

12. Ray, R.D., B.V. Sanchez and D.E. Cartwright, "Some extensions to the response method of tidal analysis applied to TOPEX/POSEIDON altimetry (abstract)," *Eos, Trans. AGU*, 75(16) Spring Meet. suppl., 108, 1994.

13. Hedin, A.E., "The atmosphere model in the region 90 to 200 km," *Adv. Space Res.*, 8(5), 9-25, 1988.

Experiences in Precision Attitude Determination with the Midcourse Space Experiment (MSX)

D.R. Haley and T.E. Strikwerda
Space Department
Johns Hopkins University
Applied Physics Laboratory
Laurel, MD 20723

Abstract

The Midcourse Space Experiment (MSX) satellite, designed, built and managed by the Applied Physics Laboratory (APL) for the Ballistic Missile Defense Office (BMDO), was launched April 26, 1996, and is still operational. The size, agility, and pointing accuracy requirements of MSX are highly unusual in their combination: launch mass of the satellite was 2713 kg.; it was designed to have the capability to point and track over a full 4π steradian range, with targets ranging from inertial stares to ballistic missiles, surrogates, ground points, and other satellites; real-time pointing accuracy was specified to be better than 0.1E, with jitter less than 9 μrad over 700 ms.; and post-flight absolute attitude knowledge was specified to be accurate to 9 μrad during all science events. All indications are that each of these requirements except the last has been met with margin.

However, as became abundantly clear very soon after launch, the accuracy of the post-flight or Definitive Attitude (DA) estimation process fell very far short of the 9 μrad requirement, to the distinct detriment of science instrument calibration and data analysis. Early efforts to remedy this inadequacy essentially concentrated on "tuning" the estimation process as originally implemented, and were demonstrably ineffective. Finally, late in 1996, it was realized that a fundamental re-evaluation of the attitude sensor suite and the DA estimation algorithms and procedures was necessary. This process has continued to the present and the results are simultaneously rewarding and maddening.

This paper presents a history and summary of our findings about the accuracy of the MSX attitude estimation process. Specific areas of discussion include:

< Definition of accuracy requirements; evaluation criteria; definition of frames of reference and differentiation among the various contributors to "absolute accuracy";
< Estimation algorithm design; data processing , reporting and quality assurance procedures;
< Inertial sensor error models and their mis-use;
< Inertial sensor requirements and initial calibration specifications; and
< Attitude system in-flight calibration requirements, experiment design and requirements, and error observability considerations.

An overview of the achievable pointing accuracy of the MSX system is discussed, and particular recommendations are presented as to how designers of "MSX II" might avoid many of the difficulties we have come upon.

Introduction

The Midcourse Space Experiment (MSX) satellite, launched April 26, 1996, was designed, built and managed for the Ballistic Missile Defense Office (BMDO) by the Applied Physics Laboratory of Johns Hopkins University (APL/JHU), and is still operational. With a launch mass of 2713 kg. and an initial inertia tensor of

$$I_{body} = \begin{bmatrix} 1739 & 17 & -104 \\ 17 & 5660 & 35 \\ -104 & 35 & 5538 \end{bmatrix} \mathrm{kg\,m^2} \quad ,$$

MSX is the largest and most complex spacecraft built by APL. References [1,2], special issues of the Johns Hopkins APL Technical Digest devoted to the MSX mission, describe the vehicle, its mission and history, and the individual science instruments in some detail. Figure 1. is a conceptual view of the satellite and its major experimental instruments, and gives some idea of the vehicle's physical complexity.

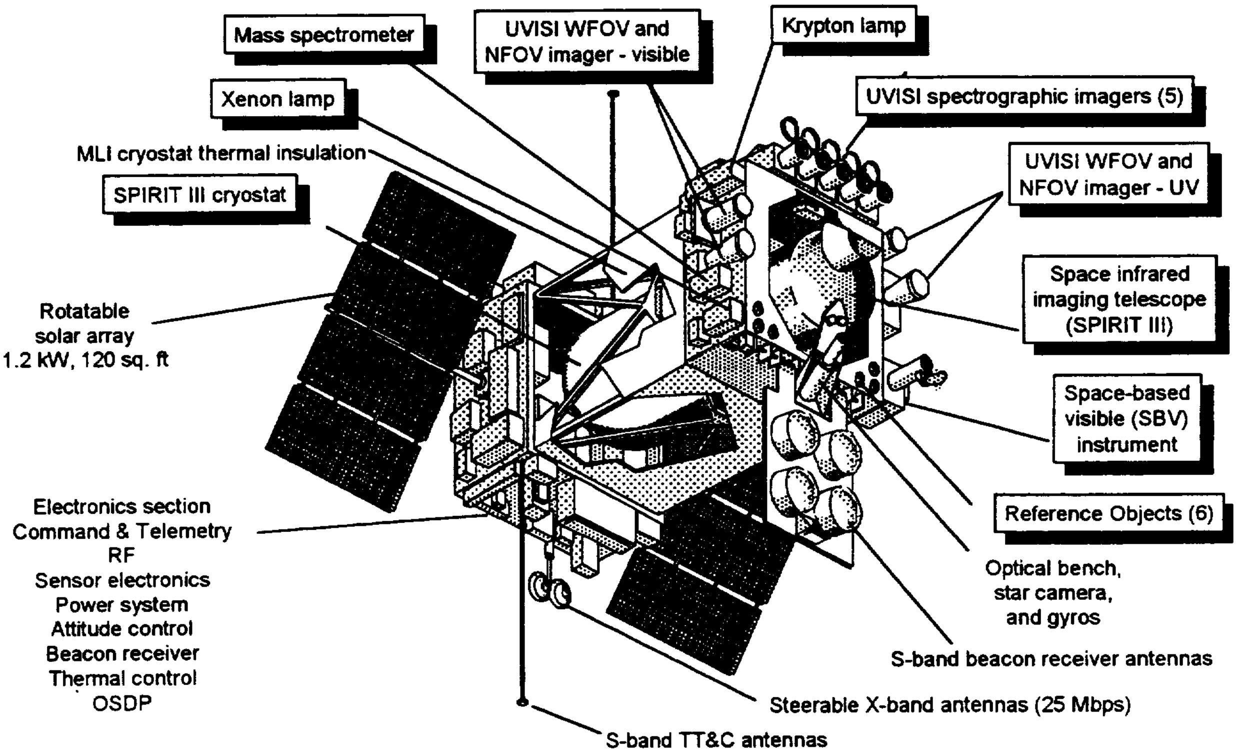

Mass spectrometer
UVISI WFOV and NFOV imager - visible
Krypton lamp
Xenon lamp
UVISI spectrographic imagers (5)
MLI cryostat thermal insulation
SPIRIT III cryostat
UVISI WFOV and NFOV imager - UV
Space infrared imaging telescope (SPIRIT III)
Rotatable solar array 1.2 kW, 120 sq. ft
Space-based visible (SBV) instrument
Reference Objects (6)
Electronics section
Command & Telemetry
RF
Sensor electronics
Power system
Attitude control
Beacon receiver
Thermal control
OSDP
Optical bench, star camera, and gyros
S-band beacon receiver antennas
Steerable X-band antennas (25 Mbps)
S-band TT&C antennas

Figure 1. MSX Spacecraft and Major Instrument Locations

In order to meet its mission science objectives, rather demanding requirements on vehicle attitude agility and pointing accuracy were specified. Real-time requirements for estimation and control may be basically summarized as:

1. Point the vehicle to the commanded attitude to within 0.1 deg.;
2. Maintain pointing stability to within 9 μrad over 700 msec.;
3. Track at commanded rates of up to 1.5 deg/sec and commanded angular accelerations of up to 0.03 deg/sec^2.

The first of these requirements is to ensure that the observed target is near the center of the field of view of the science instruments. The second is derived from pixel sizes and integration times of the instruments. And the third is necessary to track ballistic missiles and their surrogates, as well as satellites and other moving objects. By all measures, the on-board guidance and control system has met all these requirements with ample margin.

Then, for reconstruction post-event, it was specified that attitude should be estimated to an accuracy of 9 μrad, each axis, with output at 20 Hz. ("definitive attitude files" or DAF), and it was here that the real problems began. First, there was no agreement within the MSX community as to what "9 μrad" accuracy was to mean, whether it was "1-σ", "3-σ", or "do not exceed". Second, it was never agreed to which frame this accuracy applied, whether it was a "Spacecraft Fiducial Frame" or the individual frames of reference of each of the three primary instruments (SPIRIT III, UVISI, and SBV. See References [1,2] for descriptions of the instruments.), and the means of determining the relative alignments of each instrument to any reference frame were not well thought out and demonstrated to be robust. Finally and perhaps most important, several analytic assumptions regarding attitude kinematics, dynamics and estimation were made which ultimately rendered meeting any "9 μrad" requirement virtually impossible.

Attitude Sensors

The attitude sensor suite of MSX is conceptually divided into "coarse sensors", consisting of:

- Two EDO/Barnes IR Horizon Scanners;
- Five Digital Solar Aspect Detector (DSAD) Sun Sensors, provided by Adcole Corp.; and
- A Three-Axis Vector Magnetometer, provided by Schonstedt Instrument Co.,

and "fine sensors", consisting of:

- A single Ball 601 Star Tracker, and
- Dual Independent Inertial Reference Units, provided by Honeywell and based on GG1320 gyros.

Generally speaking, the coarse sensors are relevant only for attitude acquisition and "SAFE" mode operations, and will not be discussed further here (see Reference [3] for a discussion of spacecraft modes).

The Ball 601 Star Tracker is a CCD-based device, capable of tracking up to five individual stars within its 8Ex8E field of view (FOV). Calculated centroid positions of the tracked stars are output to the attitude system at approximately 9 Hz. in tracker coordinates, and the observed pattern is compared to an on-board star catalog of approximately 9000 stars for inertial attitude computation.

This "single-frame" attitude estimate is computed with the QUEST algorithm (Reference [4]), and provides the fundamental whole-value attitude measurement for both the on-board and definitive attitude computations. Since these measurements are made in the tracker coordinate system, the estimated attitude quaternion will represent the rotation from inertial to tracker coordinates, represented as qI2ST. Then MSX body attitude in inertial space, qI2B, is computed using the measured attitude of the star tracker in a body-fixed (B) frame, denoted qST2B, so that qI2B / qI2ST θ qST2B, where 'θ' denotes quaternion composition (Note that this order of composition may be non-standard. The notation throughout this paper is consistent: "qA2B θ qB2C = qA2C".). Because the tracker represents our most fundamental measurement of whole-value attitude, the star tracker frame is taken as the fiducial frame throughout this paper (equivalently, all comparisons may be made in the body (B) frame with qST2B fixed).

Based on specifications provided by Ball Aerospace and the acceptance data package for the MSX tracker, it appears that the centroid location error for each tracked star should be described by a random variable with standard deviation $\Phi \approx 15$ μrad *when the true centroid is taken as randomly positioned in the FOV.* Extensive analyses and simulations (Reference [5]) have indicated that under this assumption, the QUEST algorithm generates single frame attitude estimates with an accuracy of approximately 150 μrad about the tracker boresight and approximately 7 μrad about the tracker transverse axes.

The redundant gyro packages supplied by Honeywell each consist of a nominally orthogonal triad of GG1320 ring laser gyroscopes (RLGs) and a Mil-Std 1750 processor. Output to the attitude system is at 20 Hz., and the primary product is a whole value quaternion representing the transformation from "gyro case at turn-on to current gyro case", qG02G. From sequential quaternions, we may derive measurements of differential body motion at 20 Hz., represented in gyro case coordinates,)2$_{gyro}$, and in body coordinates,)2$_{body}$ = TG2B)2$_{gyro}$ where TG2B is the nominally constant transformation from gyro to body coordinates.

At this point, little can be determined definitively about the specified accuracy required of the gyro units. It appears that Honeywell believed, based on test data or modeling that the residual uncompensated bias drift (BD) rates were not greater that 0.1°/hr., each axis, and that the angle random walk errors (ARW) were not greater than 0.007°/hr$^{1/2}$ While ARW cannot be quantified on-orbit, we can observe long term drift data. Some very long term estimates of BD have indicated that this error is of the order 0.002 °/hr., while shorter term estimates indicate that over periods of up to two hours drift rates on the order of 0.5 °/hr. or more may be observed. Data regarding residual internal alignment errors are not readily available, but some events provide observability to a limited set of these terms. This will be discussed in some detail later.

The placement of the gyro packages and the star tracker are illustrated in Figure 2. The coordinate axes shown there are the spacecraft body-fixed axes, in which pointing accuracy discussions are framed. All of the science instruments have their boresight axes nominally aligned with the spacecraft +X axis.

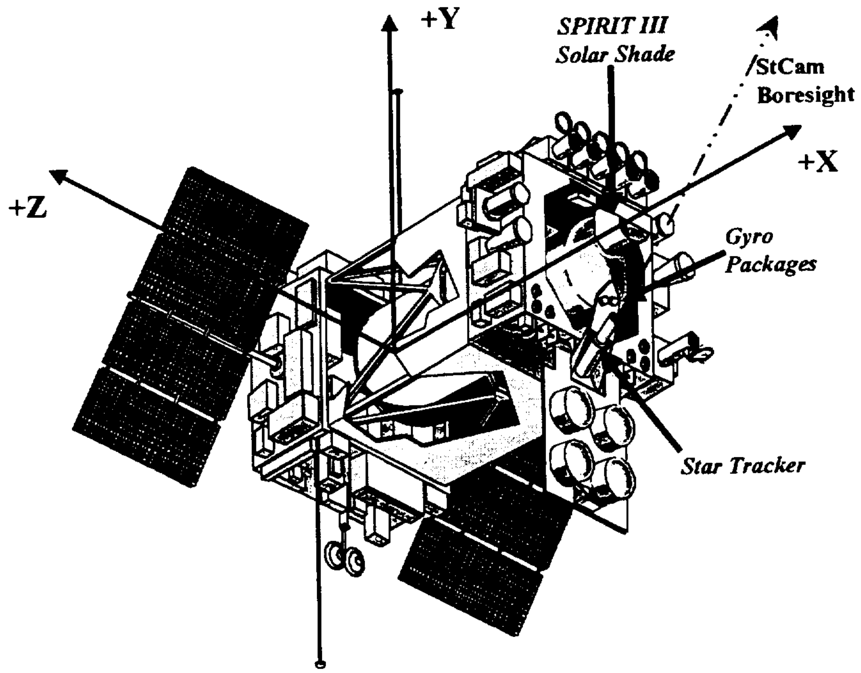


Figure 2. Attitude Sensor Locations and MSX Body Coordinate Axes

Coordinate System Definitions & Alignment Estimation

As indicated earlier, the original requirement for post-flight attitude determination for MSX was for an accuracy of 9 µrad, but with undefined statistical meaning and with undefined error contributors. For example, it was never decided whether this is a 1-σ requirement or a 3-σ requirement, or a requirement against some other statistic. And it was never decided whether this meant absolute accuracy of each instrument pointing accuracy or accuracy of a fiducial frame.

For the purposes of this paper and ongoing analyses, we have established the star tracker internal coordinates as the absolute fiducial frame of reference, and define the MSX body coordinate frame to be the tracker fiducial frame transformed by the alignment of the tracker as last measured pre-launch. All results presented here and elsewhere relative to attitude accuracy are stated in one of these (equivalent) frames.

Then the alignment of the various instruments and the gyros must be known in this frame.

While the gyros and star tracker are mounted on an optical bench and should be quite stably aligned relative to one another, there is evidence that noticeable shifts in their relative alignment do occur. And since, with the possible exception of SPIRIT III (which was supposed to be self-aligning with its autocollimator), there is little reason to expect that instrument boresights would not move relative to the tracker, the *necessity of measuring gyro and instrument alignments for each event* seems clear.

However, relatively little effort was devoted to this task and very little consideration of observability was given to the design of alignment estimation events. The result has been that relative precision has effectively replaced absolute accuracy in any measure of system performance.

Experiences in Attitude Estimation

Since MSX was launched a seemingly unending sequence of "experiences" have been encountered in our attempts to provide accurate, adequate and believable estimates of spacecraft attitude for the science instrument investigators. These have ranged from those requiring minor fixes to discoveries of fundamental flaws in the design of the overall attitude system and from discoveries of subtle but important modeling nuances to instances of serious naivete. The following sub-sections discuss the more important of these experiences and our attempts to alleviate their effects. The ordering is largely chronological by time of encounter.

Estimation Software Design & Quality Control. The attitude estimation software for both the on-board real-time system and the post-flight definitive system were conceived as straightforward implementations of a Kalman filter, very much as described in References [6,7]. Because of computational capability limitations on-board, the real-time system was designed to operate at 2 Hz., while the DA system operates at 20 Hz. Initial detailed simulation and testing of both systems indicated completely adequate performance.

However, in implementing the DA system, one initial mistake was made in our failure to read with understanding an early paragraph in Reference [6], and our attempt to implement a too-detailed dynamic model, treating gyro rates as measurements. Quoting from [6],

> "The dynamic equations for the spacecraft attitude pose many difficulties in the filter modeling. In particular, the external torques and the distribution of momentum internally due to the use of rotating or rastering instruments lead to significant uncertainties in the modeling. For autonomous spacecraft the use of inertial reference units as a model replacement permits the circumvention of these problems. In this representation, the angular velocity of the spacecraft is obtained from the gyro data. The kinematic equations are used to obtain the attitude state and this is augmented by means of additional state vector components for the gyro biases. Thus, gyro data are not treated as observations and the gyro noise appears as state noise rather than as observation noise."

Noting and heeding this paragraph led to the first significant improvement in the DA system. For a number of reasons, the real-time system was always implemented treating gyro data as kinematic truth.

In addition to the difficulties brought on by early design problems in the filter, the procedures for generating attitude files left much to be desired. It was presumed by program management that precision trajectory generation could be implemented as an automated process. The software was initially designed to run nearly autonomously without the intervention of any person who actually could tell whether the results were valid or met any tests of reasonableness. This was realized fairly early on by some of the staff but management has only recently accepted the necessity of intervention by experienced analysts in this sort of "data-intensive" estimation.

This basic extended Kalman filter software design was then used, modified, tuned, tested and extended through several versions as we attempted to generate attitude history files that were believable. Eventually, after "Version 5" of the program, a more fundamental look at error models for the star tracker was undertaken.

Star Tracker Error Models. In all of our initial design work for the MSX DA system, we relied on the basic assumption that errors in the star tracker were fundamentally white gaussian noise, thus reducible by filtering. It was discovered later that this assumption is invalid, but not until it had serious effects on definitive attitude software design.

Recall that all the science instruments have their boresights nominally aligned with the spacecraft +X axis. Thus small attitude errors about this axis do not affect instrument pointing to first order. Therefore, in the case where attitude knowledge errors are not isotropic, one should design the attitude estimation system so that the least accurate knowledge is +X. In the case where absolute pointing is derived from a single star tracker, this corresponds to mounting the tracker so that its boresight is along +X. However, many MSX experiments require pointing the instruments near the Earth limb, or even at the hard Earth. Thus, to attempt to minimize stray light interference with the tracker, its boresight was tilted about the +Z axis by 20 deg., so that it should point significantly above the horizon during earth limb tracking events. If tracker errors were white (hence reducible by filtering), this should have caused little or no degradation in pointing knowledge accuracy.

First, the tracker errors have a component which is distinctly not white noise, as was discussed in Reference [4], and in fact is a nearly deterministic function of image position on the tracker CCD. This is easily seen in Figure 3., extracted from the acceptance test report for the MSX tracker. In this test, a simulated star image was dragged slowly across the CCD, and the sinusoidal nature of the observed centroid position error is clear. The signal is at CCD pixel frequency. A further example of this sort of behavior is seen in Figure 4., derived from MSX flight data. Here five stars are tracked continuously as the vehicle moves slowly in a scan. Taking an arbitrary star as a baseline, the angular distance to the other four is plotted as a function of time. Since the stars are fixed in inertial space, the observed variation in angular distance among them must be a characteristic of star tracker errors, and it seems clear that these errors are not white noise.

The result of these error characteristics coupled with the tilt of the tracker boresight is that the error in boresight pointing knowledge becomes highly correlated with science instrument pointing, and cannot be removed. There has been much study and discussion of this phenomenon, and several means of eliminating or reducing these errors have been proposed, ranging from various forms of filtering or smoothing to modeling the errors as a time-correlated process (Reference [9], for example). But the fact at this point is undeniable that the tracker errors as represented in Figures 3. & 4. cannot be reduced by standard filtering or smoothing techniques. To fully appreciate this, one need only consider the case of an inertial stare with the tracker observing the same stars continuously. The constant component of the errors as illustrated above cannot be distinguished from the true signal, thus cannot be removed. This has been discussed further in Reference [5], where it is shown that with the MSX

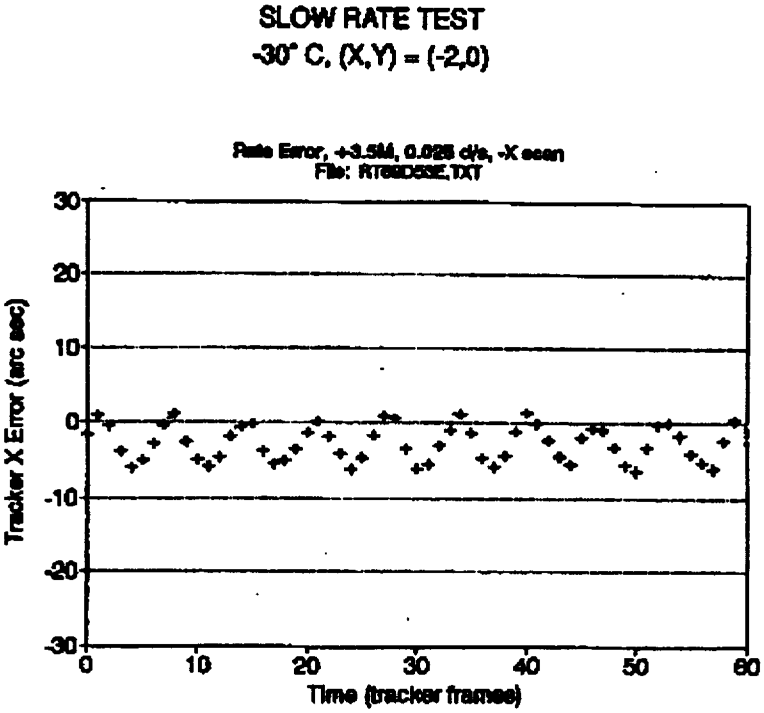


Figure 3. Typical Star Centroid Position Errors From Test Data

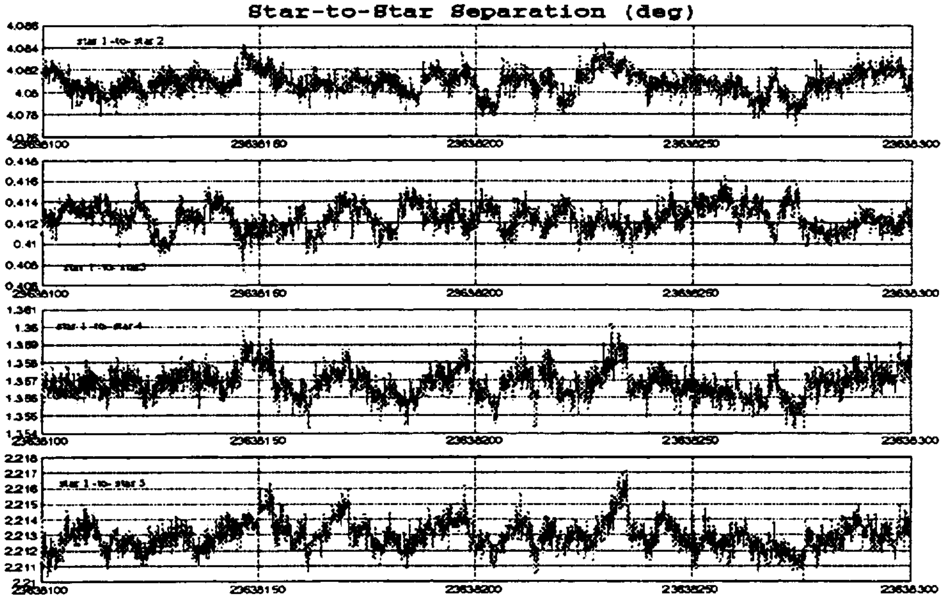


Figure 4. Observed Star-to-Star Angular Separations (MSX Flight Data)

tracker alignment, and presuming a 1-σ tracking error of 15 μrad for each of five tracked stars, the attitude knowledge error covariance is

$$P_{BODY} = \begin{bmatrix} 132^2 & -6300 & 24 \\ -6300 & 48^2 & -9 \\ 24 & -9 & 7^2 \end{bmatrix} \mu rad^2,$$

as compared to the tracker frame covariance of

$$P_{TRACKER} = \begin{bmatrix} 140^2 & 23 & 26 \\ 23 & 7^2 & 0 \\ 26 & 0 & 7^2 \end{bmatrix} \mu rad^2.$$

The introduction of errors into instrument pointing is clear, and has been observed in instrument data.

It is immediately clear that the incorporation of the information provided by a second tracker mounted with its boresight approximately orthogonal to the first will render the total single frame attitude error covariance approximately isotropic. Thus the fundamental information problem which has plagued MSX attitude estimation would have been effectively eliminated even given our assertion of the basic irreducibility of some classes of tracker errors. A second tracker was considered for MSX, but was rejected for a number of reasons. At least one of those was the lack of understanding at the time of the non-white nature of the tracker errors. It is rewarding to report that, after much programmatic tooth-gnashing, two trackers were incorporated into the design of the Thermosphere, Ionosphere, Mesosphere, Energetics and Dynamics (TIMED) satellite now under construction at APL (TIMED has an isotropic attitude knowledge requirement).

An example of tracker roll error affecting estimated attitude is seen in Figure 5. This is a plot of real data from a calibration event for the SPIRIT III instrument, where the spacecraft is commanded to rotate at –50 μrad/sec about the spacecraft +Z axis. Seen in the figure are the trajectory across the SPIRIT III focal plane as observed by SPIRIT III ('REFERENCE'), as estimated by the on-board system, as estimated from integrated gyro data, and as computed by the star tracker (note that the saw-tooth nature of the SPIRIT III has since been removed by model improvements). We note the strong similarity of trajectory among the first three trajectories immediately (the offset is alignment, which can be readily removed), and the very large variations in the star tracker single-frame solutions. The "snatches" there correspond to changes in tracked stars as the spacecraft moves, and it seems clear that no noise reduction type algorithm can reasonably remove that signal.

It was at about this point in the program that we began to understand the fundamental problem of correlation (actually causality) between star tracker roll errors and instrument pointing errors and the difficulty in attempting to filter out nearly deterministic errors. As noted above, Figure 5. indicates that the relative motion (except for alignment) of the gyro-generated trajectory and that of SPIRIT III are really quite similar. Since absolute accuracy can be calibrated out it was then decided to look at generating DA trajectories from gyro data with the star tracker used as an instrument for overall calibration of the gyros on each event.

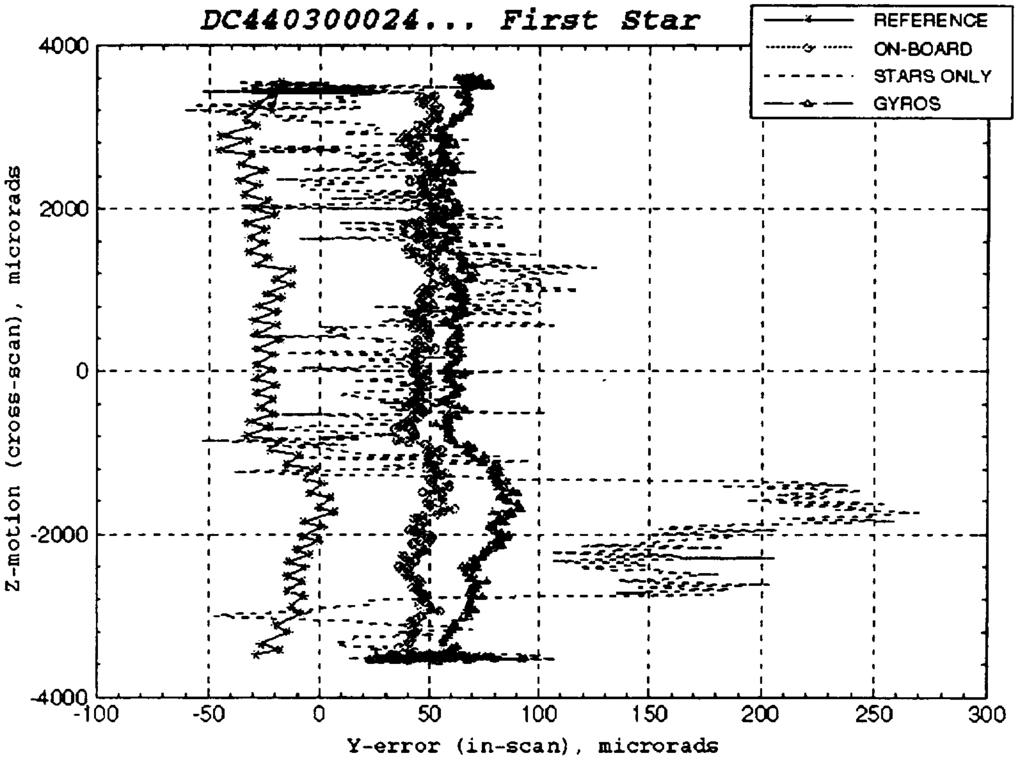


Figure 5. Comparative Error Traces in the SPIRIT III Focal Plane

We noted earlier that the Star Tracker was tilted away from the nominal boresight of the science instruments in attempt to minimize light stray intrusion when pointing near the earth limb. We should also note that this was less than totally successful, apparently due to incorrect or inadequate analysis of the tracker shade. The upshot is that in many events there are long intervals with no star tracker fixes. This not only means that the assumed continuous correction of integrated gyro errors does not take place in a recursive estimator, but also that the nominal error model for the last few available fixes is probably grossly optimistic. This difficulty is largely overcome by the batch estimator described below for the rather frequent earth limb or hard earth events where tracker data are available at the beginning and end of the tracking period.

Gyro Based Processing and Gyro Error Models. As described above the fundamental output of the MSX gyro packages is quaternions representing the rotation from the nominally orthogonal gyro case frame at initialization to the gyro case frame at the current time tag, qG02G. Then, given the orientation of the case frame at initialization and the transformation from gyro case to MSX body, we may represent vehicle attitude in inertial space by the quaternion

$$qI2B = qI2G0 \theta qG02G \theta qG2B.$$

Of these quantities, qI2G0 may be estimated to first order easily and qG2B is measured prelaunch. However, it is necessary to have estimates of these quantities more accurate than "first order" to have any chance of meeting MSX accuracy requirements.

It was also noted above that, despite the observed nature of the star tracker errors and their effect on the recursive estimator, there is every reason to believe that the errors are zero mean and essentially normal when considered over the entire CCD. Thus it seems reasonable to use the star tracker to effectively align the gyros for each event. In this formulation we seek to estimate δqO and δqA, small value adjustments to qI2B as computed above so that

$$qI2B(tracker) \equiv qI2B(true) = \delta qO \, \theta \, qI2B \, \theta \, \delta qA.$$

Based on small angle assumptions, we have formulated the least squares equations to estimate these small adjustments, and have implemented them in an iterative loop which seems to converge quite well. In addition, within this iterative loop, an adjustment is made for the appearance of a drift-like difference between the gyros and star tracker, linear in time. This is acknowledged to be an improper representation of gyro drift in the general case, but appears adequate for many events. A separate and mathematically proper formulation for estimation of gyro drift has also been formulated.

This estimator has been tested on a fairly wide variety of events with success ranging from apparently very good and understandable fits to some characteristics that were originally mystifying but now seem flagrantly and embarrassingly obvious. In any event, the algorithms have been implemented as production code and reprocessing of several hundred DCEs is in process.

As noted immediately above, a number of characteristics of gyro-to-star tracker residual plots were previously very poorly understood. Then a special request was received to process several CB05 events. These particular events are SPIRIT III DCEs, and consist essentially of three or four raster scans of back and forth motion about the spacecraft Y-axis.

The results of the star tracker to gyro fit for this event are shown in Figure 6. in the form of a time history of residuals from the fit in spacecraft X,Y,Z coordinates (units are μrad). Here we see the results of several phenomena alluded to before. First the residuals about the X-axis (top strip) are significantly larger, but apparently zero mean and looking a lot like noise. This is a result of X corresponding very closely to tracker boresight. Next, the Y-residuals (middle strip) appear to be approximately $\sin(20°) \approx 0.34$ times the X-residuals. This may be seen fairly clearly even in some of the residual patterns (not just as a statistic). Finally, the Z-residuals appear to have significantly lower noise content but a clear signal pattern. Such a signal is either not present in the X & Y plots or is masked by noise. The low noise of the Z-residuals corresponds to lack of correlation of tracker roll errors into Z.

Since this Z-signal is clearly causal, it must be the result of some unmodeled error, and there are only two sources of data. It seeming unrealistic to attribute this sort of error to the tracker, the gyros were studied a little more closely, and it was finally hypothesized that a non-orthogonality error

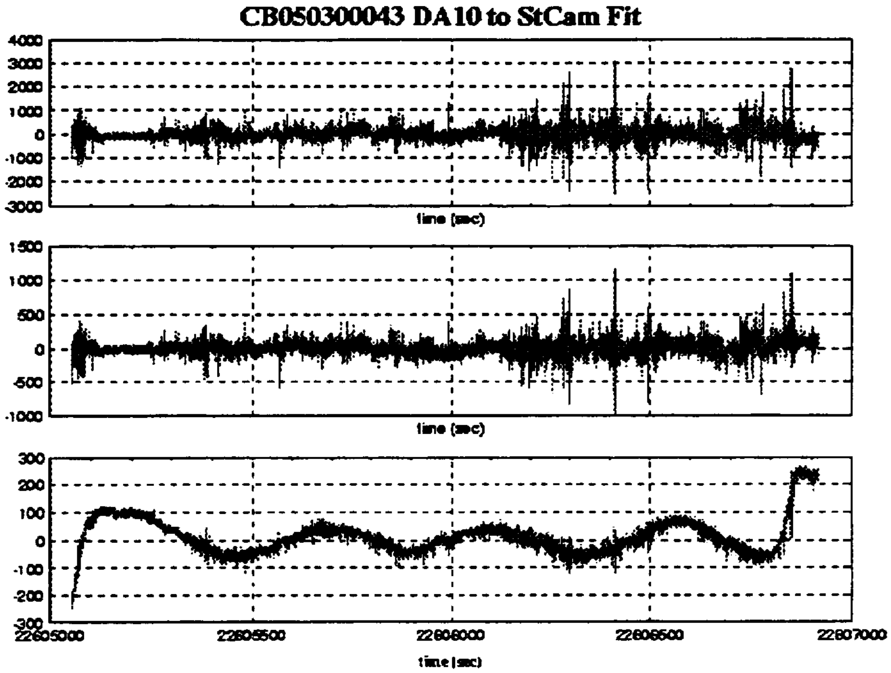


Figure 6. Star Tracker to Gyro Fit Residuals

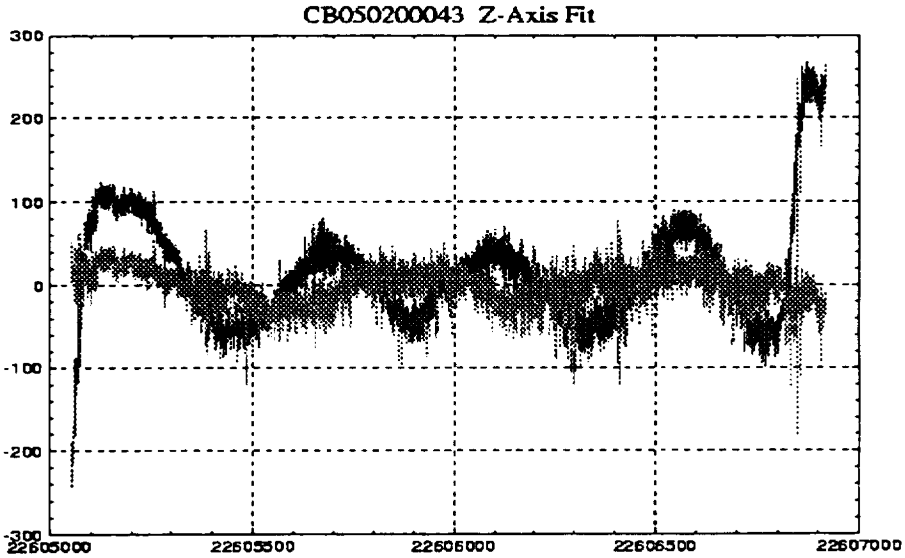


Figure 7. Z-Axis Fit With Gyro Non-Orthogonalities Estimated

in the gyro instrument mounting compensation could lead to exactly this sort of error signature in the presence of the commanded motion for this event. In retrospect, consistent, but not so obvious, residual signatures had been seen in other events for months prior to this observation.

A mathematical formulation of the error equations for instrument non-orthogonality has been formulated and implemented in our estimation software, consistent with the presentation in Reference [8]. The results of applying this estimator to this event are presented in Figure 7. (Z-axis only). The improvement in fit is immediate and dramatic. Residual error dropped from 47 μrad (1-σ) to 20 μrad, and the characteristic "tails" on the error plot (corresponding again to commanded vehicle motion) disappear. The remaining residual appears to show no pattern that might not reasonably be attributed to a combination of tracker noise and white noise and ARW in gyro data.

Summary and Conclusions

The Midcourse Space Experiment satellite has, by and large, been a significant scientific and engineering success. One particular point of difficulty, from both an engineering and scientific point of view, has been precision post-flight attitude determination. Extensive efforts have been made to meet the mission goal of 9 μrad absolute pointing accuracy, and while real improvements have been made since initial operations, the goal has clearly been missed in most instances. This shortcoming is attributed to a number of factors which are summarized here, and all contributors may generally be classified as inadequate system design, naïve untested assumptions, and inexperience in precision estimation.

Two assumptions led to critical design inadequacies in MSX. It was assumed first that the star tracker would provide uninterrupted measurements to its design accuracy specifications, and, second, that the errors in these measurements would be properly (or at least adequately) modeled as white gaussian noise. As described above the errors have a significant component which, while nearly deterministic in some senses, cannot reasonably be modeled or compensated. This leads directly to a lower bound on the accuracy information content of the measurements.

Because of the first assumption, requirements on gyro performance, particularly bias compensation and internal instrument alignment compensation were not well defined, and no effort was made to provide any on-board error calibration capability. The idea was that any accumulated gyro errors would be compensated at 9 Hz. in the DAF, and it is possible to allow significant gyro errors under such circumstances. In fact, there are many events where tracker updates are unavailable for long periods so that all navigation is necessarily via gyros only (for example, many missile tracks, all "hard-earth" tracking events, and most earth limb events). It is easy to show that gyro internal misalignment errors of 20 arcseconds or so (not a large error) can cause hundreds of microradians of pointing error on these events.

The second assumption was a major contributor to justification of omitting a second star tracker, the assumption being that errors could be adequately reduced by filtering. Also, this assumption led directly to the decision to cant the tracker away from the instrument pointing axis, thus leading to causal errors in instrument pointing. When it became clear that this modeling error could not be easily overcome by "filter tuning", and that we needed to essentially generate gyro-only trajectories, all of the same comments on gyro calibration became immediately obvious again.

While it is quite possible to estimate gyro alignment terms in real time, no effort was made to do so, and the information content in the star tracker measurements is so low about one axis that most terms are inadequately observable in the short time of a single event. Figures 6. and 7. above give one illustration of this statement not being universally true, but observations over several similar events indicate that the estimated terms vary significantly from event to event.

Finally, the program always assumed that the actual attitude determination process ("running the program") could be implemented as an automated hands-off process without intervention, editing, tuning and care by an experienced analyst. It is only very recently that this level of attention has begun to be given to DAF generation. Because of this lack of attention early on, along with the previously described problems, much of the product of the DA effort was quickly seen to be completely inadequate.

Acknowledgement

The work presented in this paper was performed under Contract HQ0006-96-D-004 in support of the Ballistic Missile Defense Office.

Bibliography

1. *Midcourse Space Experiment: Overview*, Johns Hopkins APL Technical Digest, V17, No.1, 1996.
2. *Midcourse Space Experiment: Technology*, Johns Hopkins APL Technical Digest, V17, No.2, 1996.
3. Haley, D. R., T. E. Strikwerda, J. C. Ray, H. L. Fisher, G. A. Heyler and R. T. Pham, "Performance of the MSX Guidance and Control System," AAS Rocky Mountain Guidance and Control Conference, Breckenridge, CO, 1997, Paper AAS-97-072.
4. Shuster, M. D., and S. D. Oh, "Three Axis Attitude Determination From Vector Observations," J. Guidance, Control, and Dynamics, Vol. 5, No. 1, 70-77, 1981.
5. Haley, D. R., T. E. Strikwerda, H. L. Fisher and G. A. Heyler, "Attainable Accuracy With Star Trackers", Proceedings AAS Rocky Mountain Guidance and Control Conference, Breckenridge, CO, 1998, Paper AAS-98-072.
6. Lefferts, E.J., F.L. Markley, and M.D. Shuster, "Kalman Filtering for Spacecraft Attitude Estimation," J. Guidance, Control and Dynamics, (5)5, 417-429, 1982.
7. Fisher, H.L., M. D. Shuster, and T. E. Strikwerda, "Attitude Determination for the Star Tracker Mission," Proceedings AAS/AIAA Astrodynamics Specialist Conference, Stowe, VT, 1989.
8. Britting, K. R., *Inertial Navigation Systems Analysis*, Wiley Interscience, 1971.
9. Junkins, J. L., Private Communication.

Submillimeter Wave Astronomy Satellite (SWAS) Launch and Early Orbit Support Experiences[*]

S. Kirschner, J. Sedlak, M. Challa, A. Nicholson, A. Calder, C. Sande, and D. Rohrbaugh[†]
Computer Sciences Corporation
Lanham-Seabrook, MD, USA 20706

Abstract

The Submillimeter Wave Astronomy Satellite (SWAS) was successfully launched on December 6, 1998 at 00:58 UTC. The two year mission is the fourth in the series of Small Explorer (SMEX) missions. SWAS is dedicated to the study of star formation and interstellar chemistry. SWAS was injected into a 635 km by 650 km orbit with an inclination of nearly 70 deg by an Orbital Sciences Corporation Pegasus XL launch vehicle.

The Flight Dynamics attitude and navigation teams supported all phases of the early mission. This support included orbit determination, attitude determination, real-time monitoring, and sensor calibration.

This paper reports the main results and lessons learned concerning navigation, support software, star tracker performance, magnetometer and gyroscope calibrations, and anomaly resolution. This includes information on spacecraft tip-off rates, first-day navigation problems, target acquisition anomalies, star tracker anomalies, and significant sensor improvements due to calibration efforts.

1. Introduction

The Submillimeter Wave Astronomy Satellite (SWAS) was successfully launched on December 6, 1998 (UTC). Orbital Sciences Corporation's enhanced Pegasus model XL, 3-stage, expendable launch vehicle launched SWAS into a near circular (635 km × 650 km) and high-inclination (69.96 deg) orbit. The Orbital Carrier Aircraft used to air-launch the Pegasus off the California coast is an L-1011 stationed at Vandenberg Air Force Base. Only ground sites are supporting SWAS.

The Flight Dynamics attitude and navigation teams successfully supported the launch. This support included orbit determination, attitude determination, real-time monitoring, and sensor calibration. This paper describes experiences during the launch and early mission phases. This first section gives a mission description of SWAS, a description of the spacecraft along with its attitude sensors and various attitude modes, and a tracking complement description. The next sections describe launch attempts, release and transition to a Sun-pointing attitude, software performance, and navigation results for the first day. Anomaly resolution, star tracker performance, and gyroscope and magnetometer calibrations are reviewed in the following sections. The paper concludes by describing some of the lessons learned from this launch. An overview of the mission and the Flight Dynamics support requirements can be found in Reference 1.

Mission Description

The Small Explorer (SMEX) was conceived as a low-cost program featuring a short turnaround time of typically 3 years from mission selection until launch readiness. However, because of problems with the Pegasus XL launch vehicle, SWAS launch was delayed from May 1995 to December 1998. SWAS is the fourth spacecraft to be launched in the SMEX series; the first three are the Solar, Anomalous and Magnetospheric Explorer (SAMPEX) launched in July 1992, the Fast Auroral Snapshot Explorer (FAST) launched in August 1996, and the Transition Region And Coronal Explorer (TRACE) launched in April 1998. The fifth in the series is the Wide-Field Infrared Explorer (WIRE), launched in March 1999.

[*] This work was supported by the National Aeronautics and Space Administration (NASA) / Goddard Space Flight Center (GSFC), Greenbelt, MD, Contract NAS 5-31000.

[†] Currently with AI Solutions, Inc.

NASA/GSFC, Guidance, Navigation and Control Center, *Flight Mechanics Symposium*, Greenbelt, MD USA, May 1999.

SWAS is designed for a 3-year lifetime with a 2-year science goal. Scientists are using SWAS to study molecular cloud compositions in the galactic plane by examining submillimeter spectral lines that cannot be studied using ground-based facilities. The spectral lines of interest correspond to transitions between energy levels in several chemical species, in particular: water molecule ($H_2{}^{16}O$), isotopic water ($H_2{}^{18}O$), oxygen molecule (O_2), atomic carbon (C), and isotopic carbon monoxide (^{13}CO), all of which emit in the 0.5-0.6 mm wavelength band. The data provide a mini-survey of these clouds to be used for the development of maps. The chemistry data indicate the primary means of radiative energy release from the molecular clouds, information needed for models of their temperature and pressure. This is central to understanding the gravitational collapse of molecular clouds, leading to the formation of stars and stellar systems. Additional studies include mapping local interstellar clouds, high spatial resolution studies of selected clouds, a full survey of galactic plane clouds, and examination of selected extragalactic objects.

Spacecraft Description

SWAS is a three-axis stabilized spacecraft with no thrusters. The science instrument is a single telescope operating in the submillimeter wavelength range. Figure 1 shows the SWAS spacecraft. Its attitude sensors and actuators are as follows:

- One charge-coupled device star tracker (CCDST)
- One inertial reference unit (IRU) consisting of three two-axis gyroscopes
- One digital Sun sensor (DSS)
- Six coarse Sun sensors (CSSs)
- One three-axis magnetometer (TAM) and a redundant Y-axis magnetometer
- One four-axis reaction wheel assembly (RWA)
- One three-axis magnetic torquer assembly (MTA)
- One bright object sensor (BOS)

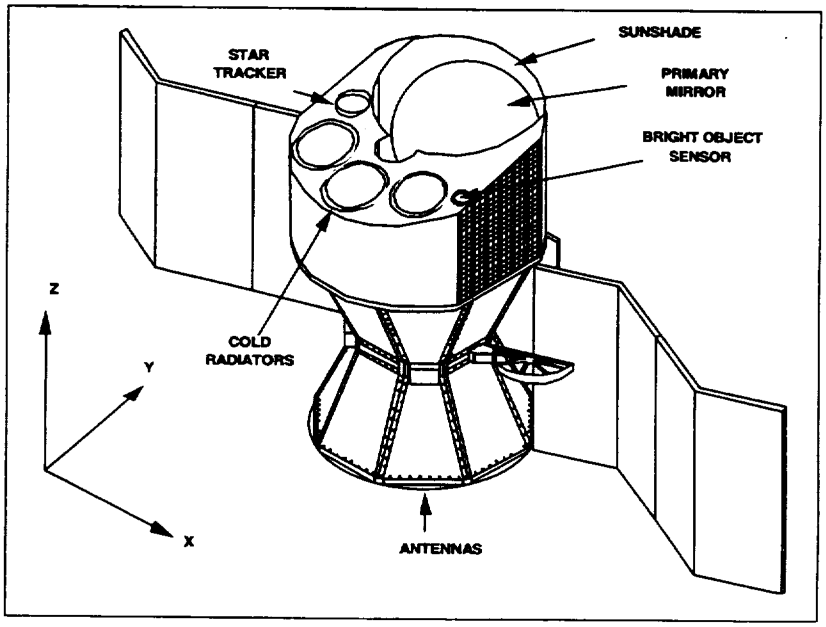


Figure 1. SWAS spacecraft.

The CCDST star tracker is the Ball Aerospace Systems Division model CT-601. It has an 8x8-deg field of view (FOV) and can track up to five stars at the same time. It is more sensitive to red stars than conventional fixed head star trackers, but this has been largely accounted for in the operational catalog of instrumental star magnitudes. The tracker is coaligned with the telescope boresight.

The IRU has three two-axis, tuned restraint inertial gyroscopes for redundancy. These gyroscopes are manufactured by Bell Textron and are of a type flown previously on sounding rockets. On some missions, the data adjustment for this type of IRU uses a different scale factor (counts to angle conversion) for rotations in the positive and negative sense. However, this asymmetry is currently thought negligible for the SWAS mission.

The BOS, a solar cell from Adcole, indicates whether the telescope pointing direction is satisfying Sun and Earth angle constraints. Other sensors and actuators share the SAMPEX heritage. Adcole manufactured the DSS and CSSs. The magnetometers and reaction wheels are from GSFC. Ithaco provided the MTA.

The SWAS personnel affiliated with Harvard's Smithsonian Astrophysical Observatory Science Operations Center (SAOSOC) in Cambridge, Massachusetts use the onboard-determined attitude for science data processing. Onboard attitude control algorithms use CCDST and IRU information with a minimum of uplinked commands (chiefly used to select targets). Because of the size of the star tracker's FOV and its ability to track multiple stars, the onboard closed-loop attitude control system (ACS) can determine accurate attitudes about the CCDST boresight (roll direction) using a single sensor. The required 3σ-control accuracy is shown in Table 1.

Table 1. SWAS attitude control accuracy requirements.

Rotation Axis	Science Control Mode		Non-Science Control Mode
	On-Target Pointing	Off-Target Pointing	
X and Y	± 57 arcsec (3σ)	± 90 arcsec (3σ)	Best Available*
Z (boresight)	± 38 arcmin (3σ)	± 38 arcmin (3σ)	Best Available*

*Dependent on control mode and available guide stars.

The control system also must keep the telescope boresight constrained to at least 75 deg from the Sun line and 35 deg from the Earth limb. Moreover, the fixed solar arrays — nominally normal to the spacecraft Y-axis — must point toward the Sun during orbit daylight to provide power. Y-axis deviations from the Sun line are restricted to rotations about the X- and Z-axes (telescope boresight) of ±15 and ±2 deg, respectively. SWAS has passive thermal control elements, but no thrusters and no known gas venting.

SWAS has five main ACS modes (listed in order of complexity):

1. Initial Sun Acquisition and Analog Safehold (momentum biased; analog control in hardware)
2. Digital Sun Point (DSP) (momentum biased; onboard computer control in software)
3. Inertial Sun Point (ISP) (boresight alternates pointing between North and South ecliptic poles; DSS and TAM data used for control)
4. Autonomous Stellar Acquisition (ASA) (boresight alternates pointing between North and South ecliptic poles; star tracker data used for control)
5. Stellar Pointing (SP) for normal operations

Ground commanding is needed to change from Analog Safehold mode to the others. The onboard system can autonomously order Analog Safehold or step down in the complexity of the control mode, as needed.

The ISP and ASA are safe "parking" modes. In these modes, the spacecraft alternates between North and South ecliptic poles, slewing roughly 180 deg about its Y-axis twice per orbit. During these rotations, the body Y-axis remains aligned with the Sun line. This keeps the spacecraft power-safe and the keeps the instrument FOV away from the Sun and Earth. If a target acquisition failure occurs while in SP, control reverts to ASA. If the spacecraft then cannot identify the pre-programmed star field near the ecliptic pole, it fails back to ISP mode where the attitude determination and control is based on the relatively coarse DSS and TAM data.

The Stellar Pointing mode can be further subdivided into Fixed Pointing, Nodding, and Mapping modes. In Nodding mode, the spacecraft remains inertially fixed as long as 45 seconds (sec) while on target, moves up to 3 deg off target (taking up to 15 sec for this motion), remains off-target for another 45 sec, and then moves back to the target. This process then repeats as often as needed for the observation. Mapping mode is similar, but the target is offset for each nod so that a map of the molecular cloud is built up over many nods. The spacecraft acquires three or four targets per orbit; hence *many* attitude maneuvers occur.

Tracking Complement Description

SWAS mission navigation is totally supported using range-rate tracking measurements. SWAS utilizes the Wallops constellation of Transportable Orbital Tracking System (TOTS) antennas currently located at Poker Flat, Alaska and Wallops Island, Virginia. These are 8-meter antennas that serve the SMEX series of missions using off-the-shelf components that have made the development and implementation of these antennas cheaper and quicker to implement than the larger standard antennas. During the first week following launch, the Deep Space Network's (DSN's) antennas located at Goldstone, Madrid, and Canberra augmented these antennas. In addition, due to the special consideration given to the paucity of early orbit tracking data, arrangements were made to obtain first-day tracking data from the Air Force C-band and North American Aerospace Defense Command (NORAD) B3 networks.

SWAS carries no thrusters, so no orbit maneuvers are possible after separation from the Pegasus XL launch vehicle. Table 2 presents the orbital requirements.

Table 2. SWAS orbital requirements.

	Position	Velocity
Predictive Ephemeris Accuracy After 14 Days	228 km (3σ)	60 m/sec (3σ)
OBC Along Track Knowledge Within 24-hr Span	50 km	60 m/sec

2. Prelaunch Activities and Early Results

The SWAS launch window for the planned launch on December 3, 1998 extended from 00:51 to 02:16 UTC. This window was selected to satisfy the constraint of keeping the spacecraft in full sunlight for at least the first five days. Launch was scheduled for 01:41 UTC near the end of the window to extend full sunlight by a day. The L-1011 took off at 00:43 UTC. Launch was aborted at 01:37 UTC and again at 01:53 UTC because the Western Range was not tracking the Pegasus. The L-1011 returned to Vandenberg Air Force Base at 02:30 UTC.

The second launch attempt was scheduled for 2 days later to allow for functional tests because the Pegasus XL had been airborne. The launch window was unchanged. Launch was planned for December 5, 1998 at 00:57 UTC, early in the window, because of predicted poor weather conditions. The launch attempt was aborted at 00:26 UTC with the L-1011 still on the ground because of the unfavorable weather conditions.

The schedule for the third launch attempt on December 6, 1998 was similar to the second. This time the launch was successful. The Pegasus was deployed from the L-1011 aircraft at 00:57:53.5 UTC, 59 minutes (min) after the L-1011 took off. SWAS separated from the Pegasus third stage at 01:09:34 UTC.

Post-Release Sun Acquisition

The initial position and velocity in geocentric inertial coordinates at the time of release from the Pegasus XL were (6435.824, -2479.835, 1292.382) km and (2.167318, 1.989943, -6.945519) km/sec, respectively. The first spacecraft telemetry was available as playback data only after the first orbit. Some of that initial telemetry for the attitude hardware is shown in Figure 2. The ACS was in Analog Safehold at release. The upper plot shows the Y-axis reaction wheel spinning up to its nominal safehold mode value of nearly 2300 revolutions per minute (rpm); the other three wheels remain commanded to zero. The spacecraft is momentum biased in this mode. The lower plot shows the Sun unit vector X- and Y-components in the SWAS body frame as measured by the DSS (the Z-component is similar to the X). The spacecraft settled into its nominal Sun pointing attitude for this mode after about 2500 sec with the Sun vector 16 deg from the spacecraft Y-axis.

Figure 3 shows components of the spacecraft angular velocity vector $\bar{\omega} \equiv (\omega_x, \omega_y, \omega_z)$. Since the gyroscopes were not powered on initially, the tip-off rates shown here could only be obtained indirectly. This was done by differentiating the spacecraft attitude history obtained from Sun and magnetic field measurements. By 2500 sec, Figure 3 shows SWAS has settled to its nominal coning motion with the transverse components ω_x (upper plot) and ω_z (not shown) exhibiting a sinusoidal behavior of amplitude 0.6 deg/sec and ω_y (lower plot) remaining constant at about 0.2 deg/sec. One feature of particular interest here is that the magnitude of the rate at release is approximately 1.5 deg/sec – well within the mission requirement of 4 deg/sec (Reference 2).

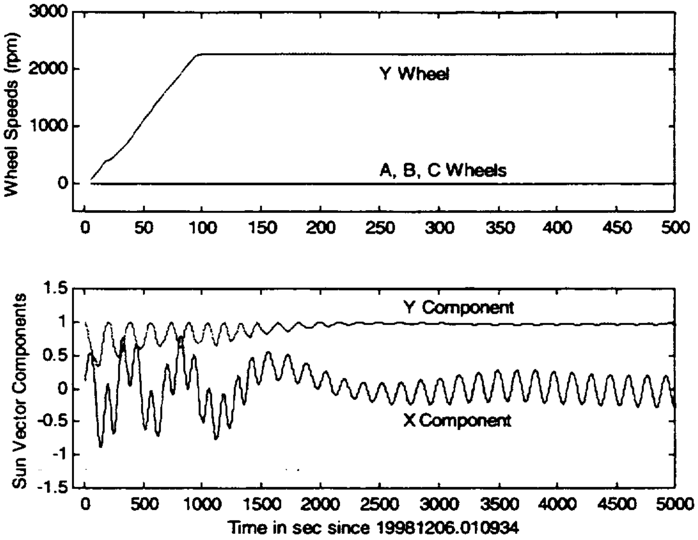


Figure 2. SWAS wheel speeds and digital Sun sensor unit vector X- and Y-components showing Sun acquisition in Analog Safehold mode just after separation.

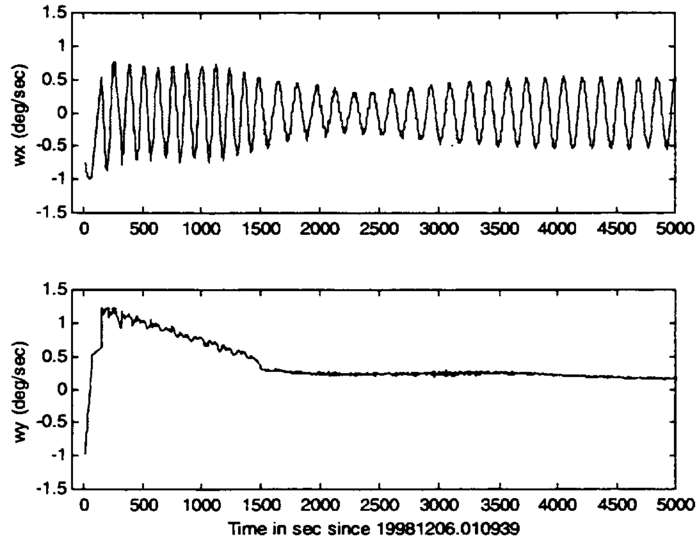


Figure 3. SWAS X-axis and Y-axis rotation rates after separation, obtained from derivative of single-frame attitude estimates.

3. Flight Dynamics Support Software

Navigation Software

The installation of navigation software in the SMEX Mission Operations Center (MOC) was intended to make the MOC a self-sufficient location in terms of providing navigation support. The suite of navigation software in the MOC is based around a core of Commercial-Off-The-Shelf (COTS) software from the Analytical Graphics Inc.'s Satellite ToolKit/Precision Orbit Determination System (STK/PODS) module, with modifications based on NASA defined formats added for end products to the tracking networks and other end users. This is the second mission supported using this technical approach (the first being TRACE and the third WIRE). The navigation software in the

MOC is a replacement for the legacy navigation software developed at GSFC and located in a second site at GSFC [the Multi-Mission Flight Dynamics facility's Mission Operations Room (MOR)].

The software in the MOC is limited in functionality during launch and real-time mission critical support. At the present time, it cannot ingest the NORAD and C-band data used in this launch. In its current configuration, the software is best suited for on-orbit operations, including orbit determination, orbital event planning, and scheduling products. During periods of rapidly changing solar activity, the software's usefulness is limited because the current configuration doesn't ingest the latest solar activity measurements. As a result, the legacy software is required during these periods to produce reliable orbit determination results. Consequently, at launch the legacy software processed the inertial guidance data and tracking data measurements from the Wallops networks, NORAD, DSN, and Air Force C-band stations. Then it provided orbital solutions and ephemerides for the MOC software to use in generating the planning and scheduling products. Improvements are planned to allow navigation support in the MOC to be independent of the legacy software in the MOR.

Attitude Software

SWAS is the first GSFC-supported mission for which all attitude support functions were carried out on PCs using the Windows NT operating system. The Attitude Determination System (ADS) was written in MATLAB. The major subsystems of the ADS are the Data Adjuster (DA), Star Identification function (STARID), Quaternion Estimator (QUEST), Batch-Least-Squares Estimator (BLS), Extended Kalman Filter (EKF), and Calibration functions (CAL). These and various other utilities were originally written for the Flight Dynamics Facility (FDF) Mainframe-to-Workstation transition, completed in 1997 (Reference 3 describes the original mainframe system). Thus these functions already have over a year of operational use in the FDF supporting six on-orbit missions. Porting the software from UNIX workstations to NT PCs required little effort since MATLAB works in both environments. Most of the ADS functions required little or no enhancement for SWAS mission requirements. The only new code development required for SWAS was a Telemetry Processor (TP), a driver and communication functions to enable real-time processing using the ADS subsystems, and two small functions for generation of calibration files for uplink to the spacecraft.

The Integration, Test, and Operations System (ITOS) unpacks the raw telemetry and produces text (non-binary) data files. The TP is a MATLAB function that reads the files, groups the data values by sensor/actuator, and passes the data on to the DA.

The Real-Time Attitude Determination System (RTADS) receives packets of data from ITOS via a TCP/IP socket. Low level functions to initialize and read data from the socket using the Winsock API were written in C. These low-level functions are called by a MATLAB real-time TP, which accumulates a user-specified number of data samples and returns to the RTADS main driver. The RTADS driver calls the DA, STARID, QUEST, and EKF functions, updates displays of results, and then calls the real-time TP for another cycle of data. Each cycle of adjusted data is concatenated to the previous cycle, so at the end of a real-time pass, the entire pass is in memory. This facilitates post-pass analysis of the real-time data. The performance of the RTADS was more than adequate to keep up with the telemetry data rate. No data was lost due to buffer overflow during any real-time pass.

The performance of the ADS in processing a 30 min span of typical playback data is summarized in Tables 3 and 4. The TP was the slowest subsystem due to the demands of reading, buffering, and parsing large volumes of text data; however, overall performance was adequate for mission support.

Table 3. Performance of Telemetry Processor.

Packet number	Data interval	Number of records	Processing time	Data contents
29	6 sec	300	3.9 sec	DSS, TAM, MTA, RWA, IRU (low data rate)
31	6 sec	300	2.2 sec	Star tracker (low data rate)
04	1 sec	1800	93.8 sec	OBC quaternion, star tracker, IRU (high data rate)

Table 4. **Performance of Data Adjuster, Star Identification, and Batch-Least-Squares Estimator.**

Subsystem	Processing time for 30 min of data
Data Adjuster	4.6 sec
Star Identification	5 sec
Batch-Least-Squares Estimator	5.9 sec

4. Navigation Results for First Day

The Pegasus final stage inserted SWAS into the final orbit on December 6, 1998 at 01:09:34 UTC. Table 5 compares the nominal orbital state, the orbital state reported via processing of the inertial guidance data, and the final orbit based on orbit determination results. The achieved orbit differed significantly from both the nominal orbit and the on-flight inertial guidance data estimate. These differences introduced relatively large positional errors for the supporting antennas during the first few hours of the mission. For previous Pegasus XL supports, the Orbital Sciences Corporation provided a post-injection assessment of the separation state using Global Positioning System (GPS) data to improve the positional estimate that has proven to be more reliable than the raw inertial guidance data. This vector was unavailable for SWAS.

Table 5. SWAS Separation and Post-Separation Vectors.

Parameter	Nominal Injection	Inertial Guidance Data – Post Injection	Orbit Determination Results
Epoch (UTC)	981206 010930.605	981206 010944.5	981206 010934.0
Semimajor Axis (km)	7045.7416	7024.8267	7028.6309
Eccentricity	0.004587	0.001279	0.001787
Inclination (deg)	69.9929	69.9593	69.9140
Right Ascension of Ascending Node (deg)	162.8101	162.8107	162.8387
Argument of Perigee (deg)	166.4561	152.2354	145.3064
Mean Anomaly (deg)	2.12077	17.0958	23.3095

Table 6 summarizes the tracking data used during the first day for SWAS. Poker Flat provided the only good tracking pass during the first few hours of the mission. Generally, orbit determination results for a new mission are not reliable for a ground-based tracking schema until the second pass. The second good pass did not occur until 8 hours after launch. By this time, the positional difference error for the supporting antennas had grown to nearly 100 km. Normal guidelines employed by navigation personnel are to try to keep positional differences under 35 km. Acquisition of the spacecraft was not interrupted due to this positional error.

Ultimately, the C-band and NORAD B3 tracking measurements were not useful during the first few hours of the SWAS mission to fill the holes in the DSN and Wallops network tracking coverage. To date, the passive tracking (C-band and NORAD) data types have proven not to be useful in the first few hours following Pegasus-based launches, perhaps due to the relative radar signatures of the Pegasus final stage and the spacecraft body itself. The TRACE and the Student Nitric Oxide Explorer (SNOE) spacecraft are other examples of this problem from 1998 launches. NORAD did correctly switch over the identification of SWAS to the proper object approximately 8 hours after launch. Caution should be given in the future for using these data types for Pegasus-based launches. However, these data types will continue to be sought for new missions until the era of GPS-based launch support is prevalent everywhere, especially for those missions where early orbit coverage by ground based antennas is spotty. These data types have been used with success for other launch vehicles and payloads, notably National Oceanic and Atmospheric Administration (NOAA)-K, which launched in May 1998. It should be noted that NORAD tracking data is generally delivered after a several hour delay to the MOR.

The early mission support for SWAS was further complicated by the need to assess whether an observation bias was needed in computation of the range-rate measurements from the TOTS antennas. Input to the TOTS receiver is down-converted, so subtracting 1980 MHz from the S-band downlink frequency determines the tuning frequency. The receiver can only be tuned in 100 kHz steps, which causes the range-rate data to have an observation bias. The range-rate bias is applied in the orbit determination measurement processing. The validation of whether to apply this bias is made on orbit to assess whether any other compensation for the bias has been made. For SWAS, it was determined that a 244.97 cm/sec bias must be applied using the Wallops WT3S data but no bias is needed using the Poker Flat WT1S data.

Table 6. SWAS tracking data for early orbit period on December 6, 1998.

Station (Antenna) & Location	Data Type	AOS HH:MM:SS	LOS HH:MM:SS	Data Quality	Maximum Elevation (deg)
WT1S Poker Flat AK	SRE USB30: Angles & Range-Rate	02:27:40	02:39:30	Good	21
KPTQ Kaena Point HI	C-band: Range & Angles	02:35:12	02:47:18	Bad (wrong object)	13
BELU	NORAD: Range & Angles	02:38:08	02:43:06	Bad (wrong object) & too low in elev.; data not received in real-time	4
KPTU Kaena Point HI	NORAD: Range & Angles	02:41:40	02:43:10	Bad (wrong object); data not received in real-time	13
CLAU Clear AK	NORAD: Range & Angles	04:12:17	04:12:40	Bad (wrong object) & too low in elev.; data not received in real-time	4
KPTQ Kaena Point HI	C-band: Range & Angles	04:13:18	04:28:48	Bad (wrong object); data not received in real-time	35
KPTU Kaena Point HI	NORAD: Range & Angles	04:21:30	04:23:00	Bad (wrong object); data not received in real-time	35
ASCU Ascension Island	NORAD: Range & Angles	05:12:12	05:15:21	Bad (wrong object) & too low in elev.; data not received in real-time	6
FY4U Fylingdales Eng.	NORAD: Range & Angles	05:28:37	05:33:57	Bad (wrong object) & too low in elev.; data not received in real-time	6
DS66 Madrid Spain	SRE USB85: Angles & Range-Rate	06:59:05	07:11:31	Bad (frequency shift mid-pass)	N/A
DS46 Canberra Aus.	SRE USB85: Angles & Range-Rate	07:47:37	07:57:04	Bad (ground station coherence problem until last few minutes of pass)	N/A
FY4U Fylingdales Eng.	NORAD: Range & Angles	08:52:03	08:55:43	Good; data not received in real-time	24
DS46 Canberra Aus.	SRE USB85: Angles & Range-Rate	09:28:10	09:37:10	Good	34
WT3S Wallops Island VA	SRE USB30: Angles & Range-Rate	10:13:20	10:23:40	Good	8
FY4U Fylingdales Eng.	NORAD: Range & Angles	10:27:00	10:35:50	Good; data not received in real-time	15
THUU Thule Greenland	NORAD: Range & Angles	10:27:24	10:33:47	Good but most data too low; data not received in real-time	6

Note: NORAD data was generally received several hours after the real-time event.

5. Anomaly Resolution

The SWAS mission proceeded smoothly during the first weeks after launch. Besides the expected minor troubles with data formats and the new software platform, there were some problems involving onboard systems and hardware that were potentially more serious. The spacecraft is designed to drop back to a lower control mode whenever it fails to acquire the targeted guide stars. When this occurs, the spacecraft leaves the planned timeline and opportunities for gathering science data may be lost, so resolution of control anomalies has a high priority. (The anomalies described here were solved by the joint efforts of Flight Dynamics personnel, the Flight Operations Team, the ACS engineers, and the visiting SAOSOC mission scientists working together as an extended team. The authors are not claiming or assigning credit for these efforts.)

One serious ACS problem was found when the spacecraft was commanded from ISP to ASA mode. These both are "parking" modes that are power safe and respect pointing constraints. In both modes, the spacecraft aligns the science instrument boresight near the North ecliptic pole for half of the orbit, then slews to the South ecliptic pole for the other half orbit. The two modes are distinguished by which sensors are used for onboard attitude determination. In ISP, the attitude is estimated using the Sun direction and the Earth's magnetic field vector. In ASA, the much more accurate star tracker data is used instead. When ASA mode was commanded, it was found that the star field could not be reliably identified after the 180 deg slew from one pole to the other, and the ACS would fail back to ISP mode. After reviewing plots of gyroscope and star tracker data from the playback telemetry, it was found that the spacecraft was acquiring its first guide star after the slew (the *base* star) before the motion had fully stopped. The star acquisition algorithm uses the base star to determine where in the FOV to look for the remaining four guide stars. If the spacecraft is still moving, it will not find them in the expected, small, directed search areas, and the star field acquisition will fail. The spacecraft motion in this case was due to a small attitude overshoot at the end of each slew. Once the problem was recognized, the ACS engineers were able to retune the onboard filter to remove most of the overshoot. This was accomplished by adjusting parameters involving deceleration of the rotation. This involved some trade-off with efficiency during other, smaller *nodding* maneuvers. These, in turn, were improved by adjusting the limits for switching between separate control laws for small and large maneuvers.

When the star tracker was powered on, there were cases where the electrical bus voltage limits were exceeded. This resulted in power to the star tracker being cut off, making fine pointing impossible. Hardware engineers verified with the star tracker manufacturer, Ball Aerospace, that voltages of that size also occurred on a similar test unit and were considered nominal when the tracker was used in that particular mode. The problem was resolved by changing the tolerances in the limit checks to be in line with actual behavior.

One extremely important science target that initially proved troublesome was the Orion Nebula. Star identification failed for this target repeatedly. When identification fails on a science target, the spacecraft falls back to ASA mode (that is, orientation toward the North/South ecliptic poles using star tracker control). However, before dropping back into ASA parking mode, the tracker performs a full field of view search. This yields the positions of the first five moderately bright stars. The onboard memory and computing power are not adequate to identify and use these stars, but they are crucial for ground analysis after the fact. We were able to identify these five stars using the pattern match algorithm in the STARID subsystem and readily verified that the spacecraft attitude was close to the commanded target. However, the star taken to be the base star by the onboard computer was identified by us to be the Orion Nebula itself. The combined magnitude of the bright Trapezium stars at the heart of the Nebula plus the integrated intensity of the Nebula itself add to an instrumental magnitude of 3.2, close to the expected base star magnitude. The base star is just over 0.5 deg from the Nebula while the onboard base star matching algorithm has a tolerance of 0.5 deg. The gyroscope misalignment, which had not been determined at that time, could easily have caused the Nebula to fall within the 0.5 deg window at the end of the slew to the Orion target. The simple solution was to choose an alternate base star farther from the Nebula. With this change, this target could be reliably identified. As is often the case, the solution is simple after the problem is correctly diagnosed.

Over the first few months of the mission, a number of isolated error events occurred where the CCDST briefly indicated saturation from a bright background and lost its lock on the guide stars. Fourteen events were identified and investigated through February 1, 1999. Of these, 12 were found to have occurred while SWAS was in the South Atlantic Anomaly (SAA) region. This is a region of relatively low geomagnetic field that allows a larger proton flux to impinge on the spacecraft. The other two of the 14 events occurred within 5000 km of either of the magnetic poles. The Sun, Moon, and Earth limb were checked for several of the events and could not be the source of the bright background, being too far from the CCDST boresight. The strong correlation with the SAA indicates that the isolated errors are very likely due to ionospheric charged particles interacting with the CCD or its electronics.

6. Star Tracker and SKYMAP Ground Star Catalog Performance

The performance of the SWAS CT-601 CCDST and the star identification results using the prelaunch SKYMAP ground star catalog allowed the determination of an accurate spacecraft attitude whenever the nominal complement of sensor data was available. The star identification algorithm is described in Reference 4.

The CCDST was commanded to track stars only during periods of inertial pointing due to the high slew rates required by the SWAS mission profile. During these times, the tracker reliably tracked commanded guide stars, only occasionally failing to acquire some guide stars. This failure occurred due to two distinct reasons. Early in the mission, the combination of the uncalibrated gyroscopes and attitude overshoot (even after retuning the control system) at the end of a slew could yield a large enough error to place a commanded star outside of the search field (see Section 5). This was a rare occurrence and did not degrade attitude determination since other commanded stars were successfully tracked. The other reason some fainter guide stars are missed is due to stray light interference from the bright Earth limb. This problem increased as the orbit processed to a geometry where the Earth limb is necessarily nearer to the boresight for many important science targets.

New CCDST performance information was obtained during the SWAS science instrument calibration. This calibration involved exposing the telescope (and, hence, the coaligned CCDST) to a star field that included Jupiter (instrumental magnitude approximately -2.5). The CCDST tracked the commanded guide stars reliably despite the presence of a very bright planet in the sensor FOV.

The stellar magnitudes measured by the CCDST on SWAS were compared to magnitudes measured for the same stars by the two CT-601 trackers aboard the Rossi X-Ray Timing Explorer (RXTE) spacecraft. The predicted magnitude differences from the SWAS and RXTE SKYMAP ground star catalogs and the actual observed differences are shown in Figure 4. Differences are expected and arise from two main factors: the slightly differing shapes of the sensitivity curves for the CT-601 trackers involved, and the difference in the standard star chosen to establish a referent for the magnitude systems used for SWAS and RXTE (G0V standard star for SWAS; A0V for RXTE). The predicted differences from the ground star catalogs in Figure 4 agree well with the upper curve of Figure A-5 in Appendix A of the SWAS Run Catalog Prelaunch Analysis (Reference 5). As seen in Reference 5, the offset is due to the difference in referent stars while the slope is due to the difference in tracker sensitivity curves. The measured differences have a mean of 0.33 magnitudes, while the predicted differences have a mean of 0.46 magnitudes. The amount of SWAS data available is such that, given the inherent sensor noise, the measured differences cannot yet be separated into components reliably. A more detailed comparison should be possible in the future.

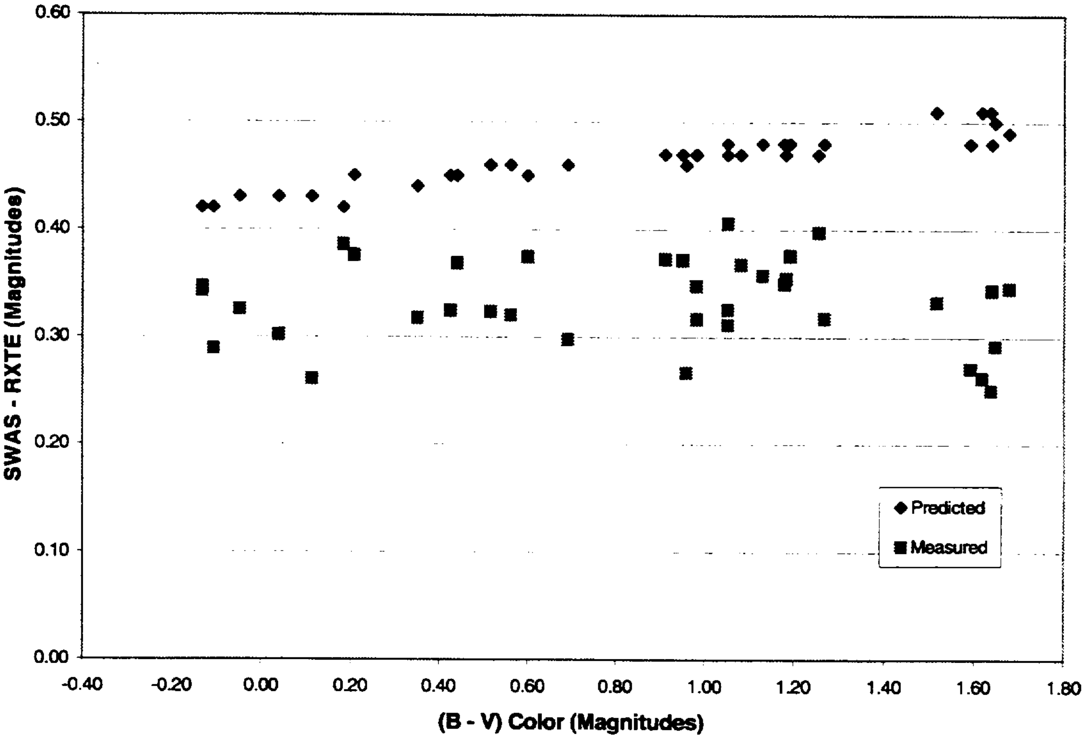


Figure 4. Predicted magnitude differences between SWAS and RXTE ground catalogs and measured differences for 37 stars observed by both spacecraft.

7. Calibration Results

Gyroscope Calibration

When the spacecraft performs a slew from one target to another, the final attitude must be close enough to the expected target to acquire the guide stars uplinked for that FOV. Once the star field is identified, the attitude is determined primarily from star tracker data. After that, the rotation rates from the gyroscopes are used in the on-board Kalman filter only to maintain a running weighted average of tracker data. However, during slews, the attitude is propagated purely on gyroscope data, so good calibration is crucial for consistent target acquisition. This is especially true for large angle attitude maneuvers.

As described in Section 1, the spacecraft reverts to ASA mode whenever it fails a science target acquisition. To ensure that ASA mode would work as a safe parking mode, it was planned to perform a partial gyroscope calibration during the first few days of the mission. The intention was to improve the accuracy of the twice per orbit North/South rotations to prevent dropping back to ISP mode. During these rotations, only the Y-axis of the gyroscope is exercised. The partial calibration uses attitudes and gyroscope biases estimated both before and after a Y-axis rotation. These attitudes are compared to the attitude determined from propagation using only the gyroscope rates. Any discrepancy is attributed to Y-axis gyroscope scale factor error. This neglects errors due to misalignment, which were expected to be smaller than the scale factor errors. (With only a single rotation axis, the available information is scalar, so only a single parameter can be estimated.) The first partial calibration was performed on December 7, 1998 while still in ISP mode. It was found that the prelaunch value of the Y-axis scale factor was low by 0.5% (a fractional correction of 0.005) which amounts to about 1 deg error for a 180 deg slew. However, these results were subject to significant uncertainty due to high sensitivity to the gyroscope bias determined using the Sun and magnetometer data before and after the slew. It was decided not to uplink a change to the scale factor until the partial calibration could be repeated using star tracker data from ASA mode. This was done on December 9, 1998 when a fractional scale factor correction of 0.0002 was found. This correction was small enough that no change to the onboard scale factor was considered necessary. It actually is consistent with zero scale factor correction, falling just within one standard deviation (1σ) uncertainty due to the errors inherent to the star tracker based attitude and gyroscope bias estimates.

A full gyroscope calibration (Reference 6) requires large rotations spanning all three axes. These could not be performed without violating Sun angle constraints while the spacecraft orbit was in full sunlight. It was planned to calibrate the gyroscopes three weeks after launch, giving time for the orbit to precess to a geometry where the shadow period was long enough to perform full 90 deg slews on all axes. In the interim, a full calibration using moderate sized slews was performed on December 20, 1998. The slew angles about the X-, Y-, and Z-axes were 43, 90, and 27.9 deg, respectively, with rotations in both the positive and negative sense. These 6 rotations plus an inertial hold of 23 min were used for the calibration. Gyroscope propagation errors for these 7 intervals were minimized over the set of 12 parameters including biases, scale factors, and misalignments for each axis. The results are given in Table 7, which shows the differences between the new calibrated values and the prelaunch values. The Y-axis fractional scale factor correction of -0.00015 is consistent with the partial calibration result of +0.0002, within the latter's uncertainty. The largest error is a rotation of 265 arcsec of the Y-axis. The X- and Y-axes both have a large component of rotation about the body Z-axis equal to -254 arcsec. Note that the X- and Y-axes are part of a single 2-axis gyroscope for this IRU configuration, so their common rotation angle indicates a simple misalignment of that gyroscope as installed in the body frame rather than any nonorthogonality in the gyroscope itself. These large X-axis and Y-axis misalignments lead to a propagation error of over 0.1 deg for a 90 deg Y-axis slew. An error of this size could explain why there was difficulty acquiring some science targets prior to the calibration. As mentioned in Section 5, the initial difficulties in acquiring the Orion Nebula guide stars probably trace to this misalignment.

Table 7. Differences between prelaunch and on-orbit gyroscope parameters based on calibrations performed on December 20, 1998 and January 13, 1999.

		X	Y	Z
December 20, 1998	Fractional scale factor correction	-0.00030	-0.00015	-0.00198
	Rotation of sensitive axis (arcsec)	256	265	115
January 13, 1999	Fractional scale factor correction	-0.00025	-0.00027	-0.00198
	Rotation of sensitive axis (arcsec)	242	255	121

The full calibration was repeated on January 13, 1999 using 90 deg slews about all three body axes. Table 7 shows both the December 20 and the January 13 calibrations for comparison. The scale factor corrections agree well within the uncertainties, and the alignments of the sensitive axes also agree closely. All the results are consistent within the error tolerances.

The calibration parameters were validated by comparing gyroscope propagation using rates adjusted with prelaunch and with on-orbit calibration values. The test is to propagate the attitude through slews distinct from the calibration slews and to demonstrate that the error angles are substantially smaller using the on-orbit calibration values. Table 8 shows the results using prelaunch values and Table 9 shows the results using the on-orbit calibration values from the full 90 deg slews. The propagation errors are much smaller using the on-orbit calibration. The average of the root-sum-squares (RSS) of the errors for the three axes is 498 arcsec for the prelaunch values and 44 arcsec for the on-orbit values, a factor of 11 improvement. Expressed as an accumulated error per degree of rotation, the prelaunch error was 5.5 arcsec/deg and the post-calibration error is 0.5 arcsec/deg, on average.

Table 8. Attitude propagation errors using prelaunch gyroscope calibration.

	X-error (arcsec)	Y-error (arcsec)	Z-error (arcsec)	RSS (arcsec)
+90 deg X-axis rotation	124.8	293.5	338.4	465.0
-90 deg X-axis rotation	101.6	216.5	156.2	285.7
+90 deg Y-axis rotation	116.4	202.8	368.0	436.0
-90 deg Y-axis rotation	353.6	53.0	129.2	380.2
+90 deg Z-axis rotation	212.1	123.9	696.4	738.5
-90 deg Z-axis rotation	18.9	85.5	676.4	682.0
Average propagation errors	154.6	162.5	394.1	497.9

Table 9. Attitude propagation errors using January 13, 1999 on-orbit gyroscope calibration.

	X-error (arcsec)	Y-error (arcsec)	Z-error (arcsec)	RSS (arcsec)
+90 deg X-axis rotation	41.7	28.9	11.4	52.0
-90 deg X-axis rotation	21.3	12.6	10.0	26.7
+90 deg Y-axis rotation	13.3	15.3	1.9	20.3
-90 deg Y-axis rotation	12.0	66.8	6.5	68.2
+90 deg Z-axis rotation	6.6	12.2	80.4	81.6
-90 deg Z-axis rotation	7.2	8.4	10.6	15.3
Average propagation errors	17.0	24.0	20.1	44.0

The most important test of the calibration is to verify that the spacecraft actually performs better using the new values. A measure of this is the size of the first onboard Kalman filter update of the attitude using star tracker data at the end of an attitude maneuver (the star tracker is not used during the slew). The tracker data is heavily weighted in the filter, so the change in the onboard estimated attitude in this first update step is close to the true propagation error. Averaging this measure of the propagation error from five slews before and five slews after uplinking the December 20, 1998 calibration parameters, it was found that the mean error angle was reduced by a factor of 6 from 378 arcsec to 64 arcsec.

Magnetometer Calibration

The TAM telemetry is processed onboard SWAS using the following model (Reference 7):

$$B_{adj} = R\, B_{counts} + b - C\, D$$

where:

B_{adj} = adjusted measured magnetic field vector in the spacecraft frame

R = *diagonal* matrix of scale factors
B_{counts} = vector of TAM measurements in counts
b = bias vector
C = torquer contamination matrix
D = vector of torquer dipole moments

The TAM residuals are then defined as B_{adj} - B_{pred}, where B_{pred} is the predicted field computed using the attitude estimate and the reference magnetic field in the inertial frame.

Note that the onboard TAM model does not include possible misalignments of the magnetometer axes with respect to the spacecraft body frame. Ground calibration of the 15 components of R, C, and b in this model was accomplished using an attitude-dependent calibration algorithm developed specifically for this TAM model (Reference 8).

On Day 2 of the mission, SWAS ACS engineers determined the contamination matrix, C, using an in-flight calibration algorithm that directly examines the effects of the magnetic torquers on the TAM measurements. However, these results showed that the prelaunch value of C was adequate and no changes were effected at that time.

Flight Dynamics personnel performed a preliminary TAM calibration using one orbit of DSP mode data from Day 2 of the mission, the main feature of interest being the significant and sustained torquer activity during this mode. A minor disadvantage of this procedure is that, during the DSP mode, the TAM itself (along with the DSS) generates the attitude data necessary for the calibration. The calibration was validated by carefully monitoring the TAM residuals over a two week span after the spacecraft entered the Stellar Pointing mode, since fine attitude profiles could be generated in this mode using only CCDST and gyroscope data. This validation indicated a need for a minor additional adjustment of the Z-axis bias by 2 milliGauss (mG).

Table 10 shows the effects of calibrating the TAM. This table presents the mean and root-mean-squares (RMS) residuals for components along the spacecraft X-, Y-, and Z-axes. It also shows the statistics of the field magnitude residuals, which are the differences between the magnitudes of the measured and predicted fields (indicated as "Mag" in the table). Note that, in Table 10, the mean residuals vanish for the DSP mode data set since this data set itself was used to estimate the TAM calibration parameters. Sample residuals are also shown graphically in Figure 5. It is clear from Table 10 that calibration significantly reduced the TAM residuals; for example, the RSS residual decreases from 9.6 mG to about 2.8 mG. Thus the TAM calibration was successful in that small magnetic field residuals were obtained consistently. It should be noted that the Flight Dynamics values for the contamination matrix differed from the prelaunch values by only about 2%; the major improvements in the residuals arose from estimating the X-axis and Z-axis biases, which were of the order of 10 mG.

Table 10. Statistics of SWAS TAM residuals before and after calibrating the TAM (upper number in each cell indicates mean residual; lower number indicates RMS residual).

Data Span	ACS Mode	Residuals before calibration: Mean (mG) RMS (mG)				Residuals after calibration: Mean (mG) RMS (mG)			
		X	Y	Z	Mag	X	Y	Z	Mag
6000 sec on Day 341 starting 1207.011432	DSP	2.76 5.68	2.44 2.83	3.49 6.47	-6.72 8.49	0.0 1.48	0.0 1.32	0.0 1.25	0.22 1.48
6000 sec on Day 342 starting 1208.000028	SP	3.27 4.90	2.38 2.43	6.39 9.16	-7.09 8.88	0.39 2.54	-0.09 0.58	0.17 2.17	0.50 1.51
10000 sec on Day 344 starting 1210.101007	SP	3.67 5.76	2.22 2.35	6.23 9.25	-9.00 10.13	-1.56 2.64	0.11 0.59	0.34 1.24	0.47 1.75
4000 sec on Day 351 starting 1217.003120	SP	0.63 4.19	3.03 3.15	3.96 6.45	-4.92 6.38	-0.61 1.77	0.40 0.71	-0.08 1.27	-0.68 0.98
6000 sec on Day 353 starting 1218.235902	SP	3.20 4.91	2.94 3.09	4.50 6.83	-5.48 7.04	-0.68 2.67	0.40 0.80	-0.57 1.26	0.51 1.69

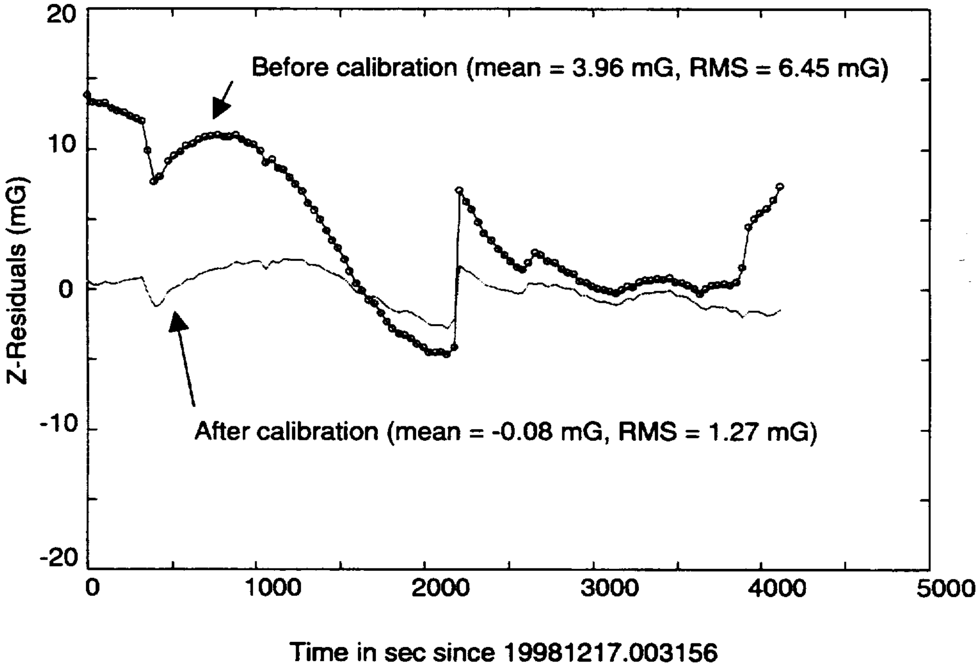


Figure 5. Typical TAM residuals showing the effects of calibration.

8. Conclusions

SWAS Flight Dynamics launch support succeeded by many different measures. The following highlights summarize the key experiences and lessons learned during early mission:

- The ground support software (also being used by the TRACE and WIRE missions) performed extremely well. Algorithms for identifying stars, determining attitude, and calibrating gyroscopes and magnetometers enabled the Flight Dynamics team to support the mission and help resolve anomalies.

- In order to improve chances of getting a stable orbit solution as soon as possible, there was the need for a variety of tracking sources during the first day of the mission. However, several bad early passes, slightly non-nominal injection, poor quality inertial guidance data, and early poor viewing geometry delayed attaining the first stable solution. Even with alternative tracking sources available, early tracking success may be limited for Pegasus-based missions.

- The SWAS star catalog, enhanced based on extensive RXTE star tracker analysis, helped improve the star identification process.

- Using attitude software that was developed in MATLAB allowed for easier modifications without disturbing configured software and the ability to perform quick analysis.

- Flight Dynamics personnel contributed to the analysis and resolution of the following spacecraft anomalies: attitude overshoot at the end of each slew; failure to target the Orion Nebula; and occurrence of CCDST single event upsets during SAA passage.

- Calibration of the gyroscopes reduced RSS average attitude residuals for a set of six 90-deg validation slews by a factor of 11. The gyroscope calibration allows the control system to maneuver more accurately to specified science targets, which, in turn, yields a higher acquisition rate of those targets.

- The magnetometer calibration reduced the TAM residuals from an RSS average of 9.6 mG to 2.8 mG. This is equivalent to improving coarse attitude solutions for contingency conditions by a factor of 3 or 4 yielding an accuracy of roughly 0.5 deg.

Many of these successes, particularly the resolution of anomalies, were greatly helped by the co-location of the Flight Dynamics launch support team with the Flight Operations Team and the ACS engineers. Co-location of personnel in the SMEX MOC was a great asset during resolution of the attitude overshoot problem, in particular. Communication among these groups was excellent and allowed for a rapid exchange of data and information regarding the anomalies as well as the analyses performed to interpret them. Solutions were formulated quickly and validated immediately after being uplinked to the spacecraft. This team effort resulted in the successful launch and early mission support of SWAS.

Acknowledgment

The authors would like to thank all of the SWAS Flight Dynamics team members for their many long and often late-night hours of effort that made this support a success.

References

1. S. Kirschner, *Submillimeter Wave Astronomy Satellite (SWAS) Flight Dynamics Support System (FDSS) Requirements Specifications, Revision 1*, Goddard Space Flight Center, 553-FDD-93/017R1UD0, April 1998

2. Orbital Sciences Corporation, *Pegasus (P20) SWAS Final Mission Analysis*, OSC Document No. TD-0619, Rev. A, November 4, 1998

3. J. Landis, et al., *Multimission Three-Axis Stabilized Spacecraft (MTASS) Flight Dynamics Support System Functional Specifications, Revision 1*, Goddard Space Flight Center, Flight Dynamics Division, 554-FDD-91/070R1UD0, September 1995

4. W. Daniel, T. Correll, and M. Anderson, "Development of a Direct Match Technique for Star Identification on the SWAS Mission," *Proceedings of the Flight Mechanics/Estimation Theory Symposium*, NASA Goddard Space Flight Center, May 1995

5. A. Miller, "SWAS Run Catalog Prelaunch Analysis," Computer Sciences Corporation, Memorandum 56830-04, September 9, 1996

6. J. Keat, *Gyro Calibration Analysis for the High Energy Astronomy Observatory-A (HEAO-A)*, Computer Sciences Corporation, CSC/TM-77/6082, June 1977

7. M. Anderson, *Small Explorer Program SWAS Attitude Control Software Algorithm Document, Build 3, Revision P*, NASA Goddard Space Flight Center, September 9, 1996

8. M. Challa, "Calibration Algorithms for Magnetometers Without Misalignments," Technical Memorandum No. CSC-86-910-20, Computer Sciences Corporation, September 1998

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 1999

3. REPORT TYPE AND DATES COVERED
Conference Publication

4. TITLE AND SUBTITLE
1999 Flight Mechanics Symposium

5. FUNDING NUMBERS
Code 453

6. AUTHOR(S)
Edited by John P. Lynch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS (ES)
Goddard Space Flight Center
Greenbelt, Maryland 20771

8. PEFORMING ORGANIZATION REPORT NUMBER
99B00051

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS (ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA/CP-1999-209235

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category: 13
Report available from the NASA Center for AeroSpace Information,
7121 Standard Drive, Hanover, MD 21076-1320. (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This conference publication includes papers and abstracts presented at the Flight Mechanics Symposium held on May 18-20, 1999. Sponsored by the Guidance, Navigation and Control Center of Goddard Space Flight Center, this symposium featured technical papers on a wide range of issues related to orbit-attitude prediction, determination, and control; attitude sensor calibration; attitude determination error analysis; attitude dynamics; and orbit decay and maneuver strategy. Government, industry, and the academic community participated in the preparation and presentation of these papers.

14. SUBJECT TERMS
Flight Mechanics, Attitude Determination, Mission Analysis, Spacecraft Dynamics, Orbit Determination

15. NUMBER OF PAGES
540

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL