HOME

Precision Formation Keeping at L2 Using the Autonomous Formation Flying Sensor

Terence H. McLoughlin[*] and Mark Campbell[†]

Cornell University, Ithaca, NY, 14853

Recent advances in formation keeping for large numbers of spacecraft using the Autonomous Formation Flying are presented. This sensor, currently under development at JPL, has been identified as a key component in future formation flying spacecraft missions. The sensor provides accurate range and bearing measurements between pairs of spacecraft using GPS technology. Previous theoretical work by the authors has focused on developing a decentralized scheduling algorithm to control the tasking of such a sensor between the relative range and bearing measurements to each node in the formation. The resulting algorithm has been modified to include switching constraints in the sensor. This paper also presents a testbed for real time validation of a sixteen-node formation based on the Stellar Imager mission. Key aspects of the simulation include minimum fuel maneuvers based on free-body dynamics and a three body propagator for simulating the formation at L2.

I. Introduction

Future NASA missions with large scale formations of spacecraft currently under study include the Micro-Arcsecond X-ray Imaging Mission (MAXIM),[1] Terrestrial Planet Finder (TPF),[2] and Stellar Imager (SI).[3] MAXIM is an X-ray interferometer composed of 33 spacecraft and will be able to image the event horizon of a black hole. TPF shall enable scientists to find and study extra-solar planets similar to our own. While several designs are currently under review, a TPF design led by Lockheed Martin is composed of 4-6 free flying spacecraft which function as an infrared interferometer.

The motivating example for this work is the Stellar Imager (SI),[3] shown in Figure 1(left). The SI mission postulates that stellar activity is key to understanding life in the universe. SI is a large, space based UV optical sparse aperture telescope/Fizeau interferometer designed to study the sun. It is designed to be flown at the Earth-Sun libration point (L2), as are other future missions such as LISA and TPF. As shown, the science of SI requires a large array of satellites in a pseudo-random placement in order to accomplish its goals. Because of the large size of the formation, studies that shed light on how requirements (fuel, precision, sensor and communication resources) change as a function of the number of satellites are extremely valuable. In addition, at the recent New Millennium Program 9 workshop,[4] the recommendations placed a high emphasis on systems level validation of formation flying technologies, as much as/more so than the development of new individual technologies. Therefore, developing and validating scalable algorithm tools is critical to the success of these missions, and is the subject of this paper.

The Autonomous Formation Flying (AFF) Sensor has been identified as a key component for precision interferometry missions[5,6] to be flown in deep space, such as those described above. It provides accurate range and bearing measurements between spacecraft based on GPS technology. An example of its usage can be found in the formation initialization sequence developed in Ref. 6. Current work by the authors in Ref. 7 has focused on developing an architecture that uses the AFF in a time-division mode to increase performance in the system.

The objective of this work is to describe a formation flying testbed at Cornell, and to use it to validate GN&C architectures for large formations of spacecraft. The Cornell real-time formation flying testbed

[*] Research Assistant, tmcloughlin@cornell.edu

[†] Associate Professor, mc288@cornell.edu

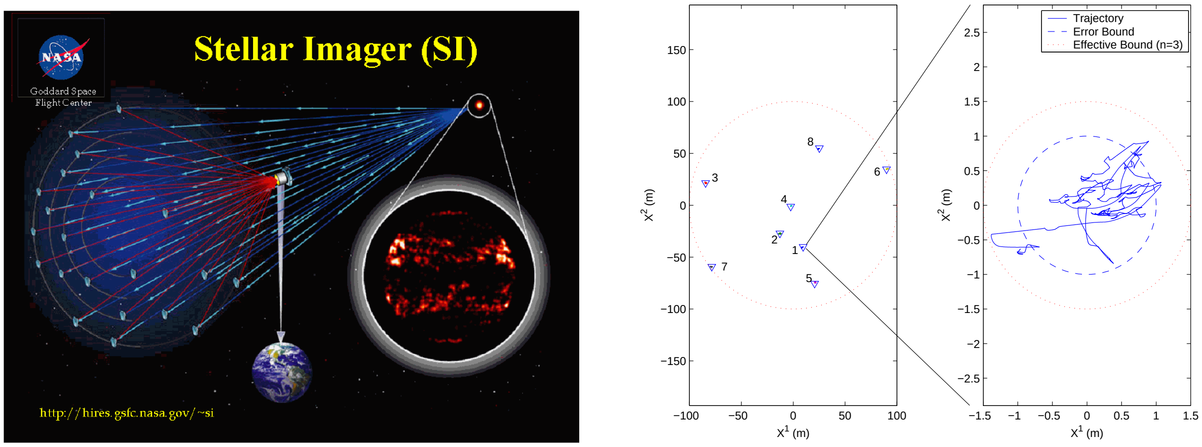

Figure 1. Left: Conceptual drawing of the Stellar Imager mission (courtesy of NASA GSFC). Right: A 2D Simulation of formation control, with a sparse aperture of 8 spacecraft (left) and close up of spacecraft 1 with error regions (right).

consists of nine Pentium PCs running real-time linux system and the ObjectAgent(OA) software.[8] The OA software allows realistic testing of system performance in the presence of slow data transfers and data packet loss by intentionally corrupting the TCP/IP Ethernet channel. Full Lagrange point dynamics are integrated to verify system performance.

The paper is outlined as follows. First, a summary of the formation flying testbed at Cornell is detailed. Next, a benchmark algorithm set is presented, based on the work detailed in Ref. 7. The algorithm set is built around the AFF sensor, and focuses on sensor architectures that can be scheduled to reduce uncertainty and fuel usage. Control, dynamics, and estimation developments are summarized.

II. Cornell Large Scale Formation Flying Testbed

The Cornell Large Scale Formation Flying Testbed consists of nine Pentium PCs. For the 16-spacecraft simulations described in this paper, two spacecraft are simulated on each of eight Pentium II PCs in the 200MHz range, and a propagator is run on a Pentium III PC. The fleet software is based on the ObjectAgent (OA)[9] framework which was designed to increase the reconfigurability, modularity and reliability of the overall control system. With its emphasis on robustly handling communication between Agents that coordinate to complete complex tasks, the OA software provides a natural framework for developing these distributed autonomous GN&C algorithms. ObjectAgent extends the classical method for writing spacecraft software by using “software agents” as the basis of the system.

A. ObjectAgent Middleware

OA is a multi-threaded architecture for distributed systems. It uses message passing for thread communication and can run on any POSIX compliant operating system. As shown in Fig. 2, there are two main components to the architecture: PostOffice and Agent. Agents are attached to PostOffices, and Agents can be subordinates to other Agents. Each entity is a separate POSIX thread. There may be any number of PostOffices on a processor, and any number of processors in the system.

The Agent is the base unit for communication, and all OA messages are passed between Agents. Agents encapsulate a set of user-defined functions which determine the behavior of the Agents. Generally, each Agent corresponds to one basic function, has inputs and outputs, and triggers one or more actions. An Agent knows its list of inputs, and outputs, and built-in functions enable it to hunt for inputs and automatically configure

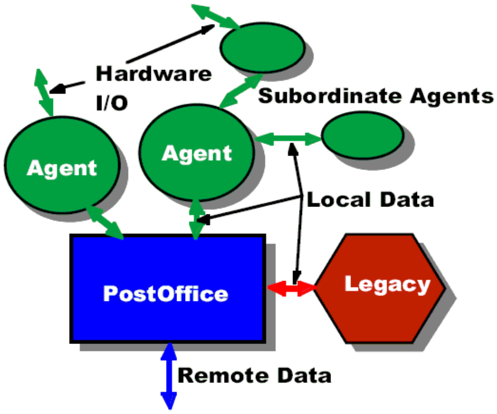

Figure 2. Diagram of a typical ObjectAgent module consisting of a PostOffice, several agents and legacy code (courtesy of Princeton Satellite Systems).

itself upon launch. In this sense, an ObjectAgent system is self-organizing.

The PostOffice enables seamless Agent communication throughout the OA system. In particular, each PostOffice manages a set of Agents, handles communication between different processors via TCP/IP or the user's choice of network protocol, and provides a framework for the dynamic creation of Agents. The PostOffice network is a fully routed network and can be reconfigured on the fly. Each link in the network can specify a separate network protocol, such as TCP/IP. Once the protocols are specified, Agent communication is transparent to the user; an Agent only needs to know the name of the data and the Agent providing or needing it.

The structure of the software and hardware environment for the testbed is shown in Figures 3 and 4. Because the GNC algorithm is a serial process, it is encapsulated in a single agent, the inputs to which are the range/bearing measurements provided by the propagator agent and thrust profiles provided by the other spacecraft in the system.[7] Each GN&C agent sends thrust commands to the propagator agent.

B. L2 Propagator

The propagator agent is based on dynamics near the sun - earth/moon L2 libration point. The L2 point is one of five points of equilibrium in the combined gravitational environment of the sun and the earth/moon

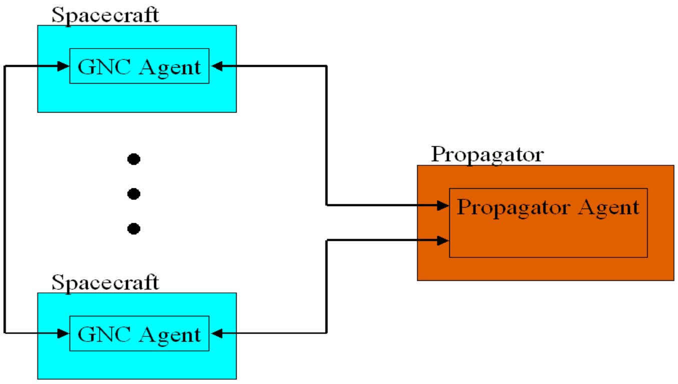

Figure 3. Agent-level diagram of the L2 simulation; postoffices not shown. The Spacecraft and Propagator may be run on any PC in the network.

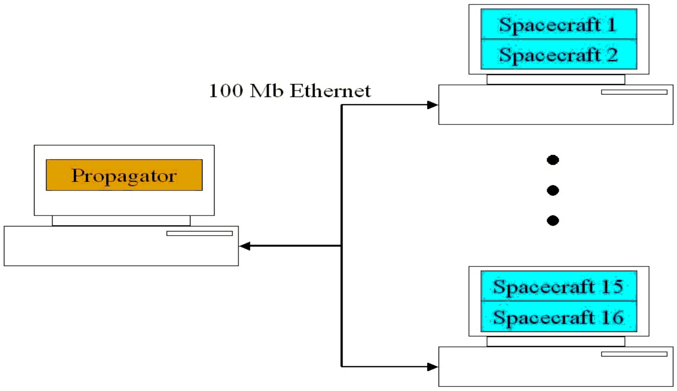

Figure 4. Hardware-level diagram of the L2 simulation consisting of nine PCs: two Spacecraft on each of eight PCs and a Propagator PC.

system. With the following assumptions: 1) the sun and the earth/moon system travel in circular orbits about their barycenter, 2) the satellite is sufficiently small as to have no effect on the motion of the sun or earth/moon, 3) no other forces affect the system, the dynamics for a satellite placed near one of these collinear points reduce to the circular restricted three-body problem (CRTBP).[10] In a coordinate frame in which the libration points are fixed, the dynamics may be written compactly with a series of Legendre polynomials:[10, 11]

$$\ddot{\vec{\rho}} = 2K\dot{\vec{\rho}} - K^2\vec{\rho} + \sum_{n=2}^{\infty} nc_nP_n\left(\frac{\rho_{\mathrm{x}}}{\|\tilde{\rho}\|}\right)\|\vec{\rho}\|^{n-2}\vec{\rho} + \sum_{n=2}^{\infty} c_nP'_n\left(\frac{\rho_{\mathrm{x}}}{\|\tilde{\rho}\|}\right)\|\vec{\rho}\|^{n-2}\left(\|\vec{\rho}\|\vec{e}_1 - \frac{\rho_{\mathrm{x}}}{\|\tilde{\rho}\|}\vec{\rho}\right) \tag{1}$$

where P_n is the n^{th} Legendre polynomial, ρ_x is the x-component of the vector $\vec{\rho}$, and:

$$K = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$c_n = (-1)^n \frac{\mu}{\|\mathrm{r}_1\|^3} + (-1)^n \frac{(1-\mu)\|\mathrm{r}_1\|^{\mathrm{n}-2}}{\|\mathrm{r}_2\|^{\mathrm{n}+1}}$$

and $\vec{r}_1$ and $\vec{r}_2$ are dimensionless vectors from the L2 point to the earth and the sun, respectively. Transformation to physical coordinates is then given from

$$\text{position (in km): } \vec{x} = \|\vec{r}_1\|\vec{\rho} \cdot 149597870$$

$$\text{velocity (in km/s): } \vec{v} = \|\vec{r}_1\|\dot{\vec{\rho}} \cdot 29.784735$$

III. Scalable GNC Architectures

The formation control problem addressed in this work is formulated as follows: given a fleet of N spacecraft, each equipped with a single AFF sensor, find a sequence of measurements that adaptively maximizes, in some sense, knowledge of the fleet. Maximizing knowledge of the fleet directly benefits state error tracking, robustness and collision avoidance, and indirectly benefits performance in terms of lowering fuel usage.[7] Several assumptions are made: 1) the AFF generates bearing measurements expressed in the inertial reference frame, 2) the AFF sensor is omni-directional, i.e. it can provide range and bearing measurements with a full

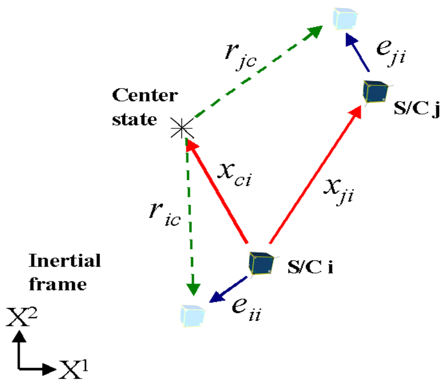

Figure 5. Virtual center state x_{ci} relative to spacecraft i, denoted by *. Also shown are reference states $r_{\cdot c}$, error states $e_{\cdot i}$ and relative state x_{ji}.

spherical field of view, 3) acquisition of signal at the AFF is instantaneous and measurements within the nominal range/bearing performance specifications of (2cm, 1arcsec) described in Ref. 5 are provided instantaneously, and 4) thrust maneuvers at each spacecraft are transmitted over a low bandwidth communication subsystem to all other spacecraft.

Over short time horizons, each spacecraft is approximately governed by second order dynamics.[12] For N spacecraft, the six element state vector (3D position and velocity) for the i-th spacecraft is $x_i(k) = \left[x_i^{(1)}(k) \quad \dot{x}_i^{(1)}(k) \quad x_i^{(2)}(k) \quad \dot{x}_i^{(2)}(k) \quad x_i^{(3)}(k) \quad \dot{x}_i^{(3)}(k)\right]$. Also, u_i is the control input, w_i is zero mean white process noise with covariance Q, m is the mass of each spacecraft (assumed to be identical across the fleet) and ΔT is the sampling time. Defining relative dynamics as $\{x,u,w\}_{ji}(k) \triangleq \{x,u,w\}_j(k) - \{x,u,w\}_i(k)\ \forall\ j \neq i$, the full system dynamics relative to spacecraft i can be written using block notation:

$$\mathbf{x}_i(k+1) = \mathbf{A}\mathbf{x}_i(k) + \mathbf{B}_\mathbf{u}\mathbf{u}_i(k) + \mathbf{B}_\mathbf{w}\mathbf{w}_i(k) \tag{2}$$

$$\{\mathbf{x},\mathbf{u},\mathbf{w}\}_i(k) \triangleq \begin{bmatrix} \vdots \\ \{x,u,w\}_{ji}(k) \\ \vdots \end{bmatrix}_{j\in\{1,\ldots,N\}\backslash i}, \tag{3}$$

$$\{\mathbf{A},\mathbf{B}_\mathbf{u},\mathbf{B}_\mathbf{w}\} \triangleq \begin{bmatrix} \{A,B_u,B_w\} & & \\ & \ddots & \\ & & \{A,B_u,B_w\} \end{bmatrix}. \tag{4}$$

Equation 3 is read as an augmented vector consisting of the $N-1$ relative vectors such that $\mathbf{x}_i \in \mathbb{R}^{6(N-1)}$ and $\{\mathbf{u}_i,\mathbf{w}_i\} \in \mathbb{R}^{3(N-1)}$. Similarly, Eq. 4 is block diagonal such that $\mathbf{A} \in \mathbb{R}^{6(N-1)\times 6(N-1)}$ and $\{\mathbf{B}_\mathbf{u},\mathbf{B}_\mathbf{w}\} \in \mathbb{R}^{6(N-1)\times 3(N-1)}$.

For formation control, a reference state for each vehicle in the formation is defined. Following Ref. 13, a *virtual center state*, x_c, is arbitrarily defined in the inertial reference frame and a local frame centered at this state is defined such that the difference between the inertial frame and the virtual center frame is only a translation. A local reference frame at each spacecraft i is similarly defined and the location of the virtual center in the i-th local frame is denoted by x_{ci}. Over relatively short time horizons the formation is allowed to drift in the inertial frame and so it is sufficient to specify the reference states in the virtual center frame. These reference states are denoted by $r_{ic} \in \mathbb{R}^6\ \forall\ i \in \{1,...,N\}$.

It is convenient to rearrange this vector for each spacecraft such that, for spacecraft i, its reference state appears first followed by the $N-1$ reference states of the remote spacecraft. Therefore,

$$\mathbf{r}_i(k) \triangleq \begin{bmatrix} r_{ic}(k) \\ \mathbf{r}_{ri}(k) \end{bmatrix}, \in \mathbb{R}^{6N} \; \mathbf{r}_{ri}(k) \triangleq \begin{bmatrix} \vdots \\ r_{jc}(k) \\ \vdots \end{bmatrix}_{j \in \{1,\ldots,N\} \setminus i} \in \mathbb{R}^{6(N-1)}. \tag{5}$$

These desired states are assumed to be known to each spacecraft in the system (as would be in a formation keeping mode).

The measurements provided by a single AFF sensor are relative range, azimuth and elevation, each corrupted with white gaussian noise. Because the spacecraft is assumed to be equipped with an inertial attitude sensor, the azimuth and elevation measurements are assumed to be made in the inertial frame and the statistical errors present in the attitude sensor are assumed to be built into the noise term. Thus, at each time step k, a measurement is made to spacecraft j,

$$z_i(k,j) = h(\mathbf{x}_i(k), j) + v_i(k), \tag{6}$$

$$h(\mathbf{x}_i(k), j) = \begin{bmatrix} R_{ji}(k) \\ \theta_{ji}(k) \\ \phi_{ji}(k) \end{bmatrix} = \begin{bmatrix} \sqrt{x_{ji}^{(1)}(k)^2 + x_{ji}^{(2)}(k)^2 + x_{ji}^{(3)}(k)^2} \\ \tan^{-1}\left(\frac{x_{ji}^{(2)}(k)}{x_{ji}^{(1)}(k)}\right) \\ \tan^{-1}\left(\frac{x_{ji}^{(3)}(k)}{(x_{ji}^{(1)}(k)^2 + x_{ji}^{(2)}(k)^2)^{\frac{1}{2}}}\right) \end{bmatrix}. \tag{7}$$

Also on each spacecraft i, estimates of the relative states, denoted by $\hat{\mathbf{x}}_i$, are maintained via an extended information filter.[14] This filter is the dual of the extended Kalman filter and maintains an information state, $\hat{\mathbf{y}}_i(k \mid l) \triangleq \mathbf{Y}_i(k \mid l)\hat{\mathbf{x}}_i(k \mid l)$, where $\mathbf{Y}_i(k \mid l)$ is the so-called information matrix,

$$\mathbf{Y}_i(k \mid l) \triangleq \left(E\left[\left(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid l)\right)\left(\mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k \mid l)\right)^T \mid Z^l\right]\right)^{-1} \tag{8}$$

The prediction step and update steps are,

$$\hat{\mathbf{y}}_i(k \mid k-1) = \mathbf{Y}_i(k \mid k-1)\left[\mathbf{A}\hat{\mathbf{x}}_i(k-1 \mid k-1) + \mathbf{B}_{\mathbf{u}}\mathbf{u}_i(k)\right], \tag{9}$$

$$\mathbf{Y}_i(k \mid k-1) = (\mathbf{A}\mathbf{Y}_i^{-1}(k \mid k-1)\mathbf{A}^T + \mathbf{Q})^{-1}, \tag{10}$$

$$\mathbf{y}_i(k \mid k) = \mathbf{y}_i(k \mid k-1) + C_j(k)^T\mathbf{R}^{-1}(k)\left[\mathbf{z}(k) - h(\hat{\mathbf{x}}_i(k \mid k-1), j) + C_j(k)\hat{\mathbf{x}}_i(k \mid k-1)\right], \tag{11}$$

$$\mathbf{Y}_i(k \mid k) = \mathbf{Y}_i(k \mid k-1) + C_j(k)^T\mathbf{R}^{-1}(k)C_j(k), \tag{12}$$

where $C_j(k)$ is the linearized measurement made to spacecraft j equal to the Jacobian of $h(\cdot, j)$ evaluated at $\hat{\mathbf{x}}_i(k \mid k-1)$. Note that the above equations can be modified if a spacecraft is equipped with $n > 1$ AFF sensors.

Finally, as described in Ref. 7, an optimal controller (minimum time or minimum fuel) based on thrust limited ($u_i^{(\{1,2,3\})} \in \{-U_{max}, 0, U_{max}\}$) propulsion is activated when the local error state e_{ii} reaches the boundary of an error ellipsoid, with switch times for the control defined as a function of starting and end points of the state error. Figure 1(right) shows an example of this controller in simulation.

A. Distributed Information Weighted Virtual Center

A virtual center state[13] is defined as the state that minimizes a weighted squared error in the formation relative states; this assumes that such a small error also minimizes the control effort required to null the error. Relative to spacecraft i, the virtual center is written as,

$$
\begin{aligned}
x_{ci}^{*} &= \arg\min_{x_{ci}} \mathbf{e}_i^T W_i \mathbf{e}_i, && (13)\\
\mathbf{e}_i &= \begin{bmatrix} e_{ii} \\ \begin{bmatrix} \vdots \\ e_{ji} \\ \vdots \end{bmatrix}_{j\in\{1,\ldots,N\}\setminus i} \end{bmatrix} \in \mathbb{R}^{4N}, && (14)\\
e_{ii} &= x_{ci} + r_{ic} \in \mathbb{R}^4, e_{ji} = x_{ci} - (x_{ji} - r_{jc}) \in \mathbb{R}^4, && (15)
\end{aligned}
$$

where W_i is a symmetric weighting matrix. The errors e_{ji} are referred to as the remote error states and e_{ii} the local error state. In Ref. 13, the center state is maintained at a single spacecraft, e.g. spacecraft $i = 1$, and W_1 is related to the fuel reserve states of each spacecraft in the fleet.

In Ref. 7, this concept was extended to allow the spacecraft to create local estimates of the virtual center. Defining first a weighting matrix W_i such that the information content of the relative state estimates $e_{ji}\ \forall\ j \in [1, N] \setminus i$ is used in the virtual center calculation, $\mathbf{Y}_i$, and a weight on the local error state, e_{ii},

$$
W_i = \begin{bmatrix} Q_i & \\ & \mathbf{Y}_i \end{bmatrix}. \qquad (16)
$$

the virtual center calculation reduces to,

$$
\begin{aligned}
\hat{x}_{ci} &= \left(Q_i + \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} \left[\mathbf{Y}_i\right]_{jk} \right)^{-1} \left(\Lambda^T \mathbf{Y}_i \left(\hat{\mathbf{x}}_i - \mathbf{r}_i \right) + Q_i r_{ic} \right), && (17)\\
\mathbf{Y}_{ci} &= \mathbf{P}_{ci}^{-1} = \left(Q_i + \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} \left[\mathbf{Y}_i\right]_{jk} \right) \left(\sum_{j=1}^{N-1} \sum_{k=1}^{N-1} \left[\mathbf{Y}_i\right]_{jk} \right)^{-1} \left(Q_i + \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} \left[\mathbf{Y}_i\right]_{jk} \right). && (18)
\end{aligned}
$$

Unobserved (poorly observed) spacecraft identically correspond to zero (or near zero) information on the unobserved states, resulting in zero (or near zero) weight on those states in the virtual center calculation. In such a case, the order of the filter, and the order of the associated center state calculation, may be reduced to eliminate the states corresponding to the unobserved or poorly observed spacecraft.

Several observations are summarized in Ref. 7. First, for noiseless sensors, each local virtual center calculation reduces to an identical state that is equivalent to the centralized solution given in Ref.[13] Second, the use of information as a weight is naturally robust, as any spacecraft that are poorly observed have small/zero corresponding entries in W_i. Finally, given range/bearing measurements, simulation results indicated that information of a resulting state estimate is roughly inversely proportional to the range to the sensed spacecraft. Thus, if two spacecraft are relatively close to each other with respect to the rest of the fleet, the weights on the corresponding error term are high. This can be advantageous for collision avoidance, as the controllers work hard to keep the spacecraft separated. However, this can be a disadvantage also, as the two spacecraft disregard the states of the rest of the fleet. When assigning the reference states of the formation, the designer should therefore ensure that the reference states of any two spacecraft not be too close relative to the fleet.

The free weights Q_i can be used to control the degree to which each spacecraft "leads" or "follows" the fleet. For example, in a two spacecraft system, as $Q_1 \to \infty$ and $Q_2 \to 0$, a leader follower scheme results where spacecraft 1 is the leader and 2 is the follower. This is especially advantageous for the case where there are failures in a node of the sensor network. One choice of Q_i is the following:

$$
Q_i = \frac{q_i}{N-1} \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} \left[\mathbf{Y}_i\right]_{jk}. \qquad (19)
$$

This term is an average of the weights on the $N-1$ remote error states (or equivalently, an average of the information of the remote relative states), weighted by q_i. Substituting Eq. 19 into 18 yields

$$\mathbf{Y}_{ci} = (1 + \frac{q_i}{N-1})^2 \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} [\mathbf{Y}_i]_{jk} \,. \tag{20}$$

The scalar q_i can be used to incorporate the local fuel reserve state by allowing q_i to approach ∞ for low fuel, and some small value or zero for high fuel. In missions such as SI, which consist of a single large hub spacecraft and $N-1$ smaller reflector spacecraft, it may be desirable to place a q_i on the hub spacecraft that is different from the reflector spacecraft, depending on the mission phase. If the formation is in the observation phase, then it may be necessary to make the hub spacecraft an effective leader. This may also be the case if the system is designed such that the hub is responsible for stabilizing the orbit of the formation about L_2. The hub is made a leader by setting $q_{hub} = \infty$ during such orbit corrections. Given the position of the hub in the inertial frame, trajectories in the inertial frame for the follower spacecraft can easily be translated via the reference states $\mathbf{r}(k)$. Conversely, the weight on the hub spacecraft may be lowered during housekeeping activities or if there is a failure in one of the AFF sensors.

B. Time Constrained AFF Sensor Scheduling

Spacecraft in formation-based missions such as SI will likely be equipped with only a single AFF sensor, and therefore, only a single range/az/el measurement is provided at each time step. Ref. 7 introduced an algorithm that switches the measurements of the AFF sensor among the spacecraft based on maximizing the resulting center state information. It was also shown that the infinite time AFF sensor scheduling problem appears to converge to a periodic solution. The solution to this sensor scheduling problem is summarized as follows.

Considering a single spacecraft (omitting the i subscripts) the finite time horizon problem is described by the utility function,

$$J_K \triangleq \frac{1}{K} \sum_{k=0}^{K} f\left(\mathbf{Y}(k)\right), \tag{21}$$

where $\mathbf{Y}(k)$ is the updated information matrix which evolves according to the information Riccati equation

$$\mathbf{Y}(k+1) = \left(\mathbf{A}\mathbf{Y}(k)^{-1}\mathbf{A}^T + \mathbf{Q}\right)^{-1} + C(k)^T R(k)^{-1} C(k). \tag{22}$$

The measurement matrices, $C(k)$, are selected from the finite measurement set, $C(k) \in \{C_j \mid j \in N_i\}$, with corresponding measurement noise covariances $R(k) \in \{R_j \mid j \in N_i\}$ where N_i is an index set, via the measurement control law $\boldsymbol{\mu}_K = \mu(0), \mu(1), ..., \mu(K)$ such that $\mu(k) \in N_i$ and $C(k) = C_{\mu(k)}$.

For the fleet estimation problem outlined at the beginning of Section III, the index set is the set of remote spacecraft measurements, $N_i = \{1, ..., N\} \setminus i$. Although the range/bearing measurement in Eq. 7 is nonlinear and thus the linear approximation $C_j(k) = \nabla h_x(j)$ evaluated at the relative state estimate at time k is time varying, it is assumed that, in the presence of a controller, the vehicles maintain formation and thus the Jacobian of the measurement function is evaluated at the formation reference states. The resulting scheduling algorithm is therefore appropriate only for fixed formations with constant reference states, $\mathbf{r}$.

For a given periodic measurement sequence

$$\boldsymbol{\mu}_\infty = \{\bar{\boldsymbol{\mu}}_T, \bar{\boldsymbol{\mu}}_T, ...\},\ \bar{\boldsymbol{\mu}}_T = \{\bar{\mu}(0), \bar{\mu}(1), ..., \bar{\mu}(T-1)\}, \tag{23}$$

where T is the period of the sequence, the resulting utility is

$$J_\infty = \lim_{K\to\infty} J_K = \frac{1}{T} \sum_{k=0}^{T-1} f(\hat{\mathbf{Y}}(k)), \tag{24}$$

where $\hat{\mathbf{Y}}(k)$ is the steady state periodic information matrix which is found by solving a $4(N-1)$th order periodic Riccati equation,[15] as described in Ref. 7.

Also in Ref. 7, an integer gradient search algorithm was developed and used, with the utility defined in Equation 24, where $f(\mathbf{Y}(k)) = \text{trace}\mathbf{Y}_{ci}(k)$ to solve for the optimal measurement policy for a given period T. Summarizing the results/insights from this scheduling approach: 1) it is better to take a single measurement to a spacecraft and quickly switch to another as rapidly as the hardware allows, 2) it is better to use a constant, deterministic sensor schedule, rather than randomly select from a distribution as in Ref. 16, 3) the utility of a set of sequences does not necessarily increase with T and 4) spacecraft that are closer are sensed more frequently than those far away.

The AFF sensor may not be able to switch at every timestep, therefore the scheduling algorithm has been extended to allow for such time constraints. The algorithm has been modified to produce the optimal measurement schedule based on two parameters: T, the integer number of timesteps in the periodic sequence, and N, the minimum switching time of the sensor, also in timesteps, such that T/N is also an integer. The resulting measurement sequences for a set of $\{T, N\}$ are shown in Figure 6 (left), and the resulting utility is shown in Figure 6 (right). The results show that for large $\{T, N\}$ the resulting optimal sequence is similar to the sequence for $\{T/N, 1\}$ with each single measurement repeated N times. This is useful for improving the speed of the algorithm since the speed of the algorithm decreases with increasing T. Finally, note that although the mean of the utility does not appear to decrease with increasing N, the mean will eventually decrease as N gets very large.[7] As N approaches ∞ only a single spacecraft may be observed and the information of the state estimates of the other spacecraft goes to zero. The resulting utility will be much less than the utilities in Figure 6 (right).

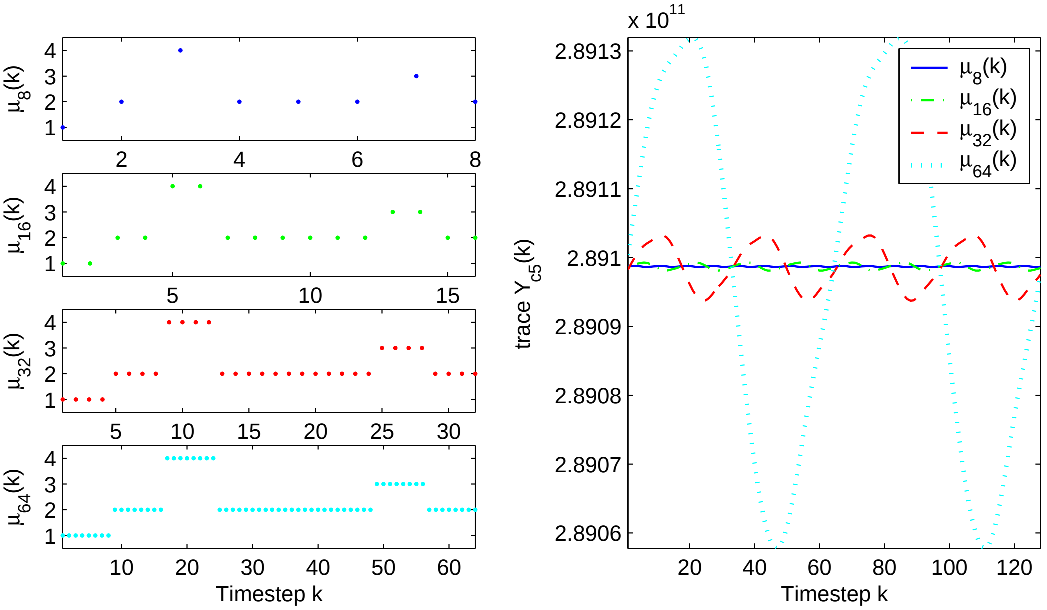

Figure 6. Resulting measurement sequences for a five spacecraft system; spacecraft 5 takes measurements to spacecraft 1-4. $T/N = 8,\ T = \{8, 16, 32, 64\}$.

C. Comparison of Centralized, Decentralized, Switching Architectures

Results of a 2D simulation of the formation shown in Figure 1 (right) are shown in Figure 7. Free body dynamics are simulated with differing constant biases at each spacecraft in the range of 20-25 μN and a small random white noise component with a standard deviation of 1μN in X^1 and X^2. Each spacecraft is equipped with $1m$N thrusters in X^1 and X^2, and each AFF sensor provide measurements at a rate of 0.25 Hz. The controller error ellipse is defined such that $e^{(1)}_{max} = e^{(2)}_{max} = 1$m and $\dot{e}^{(1)}_{max} = \dot{e}^{(2)}_{max} = 0$. Several cases are tested:

***n* > 1 Unweighted** Each spacecraft is equipped with n AFF sensors that run concurrently and provide measurements to n spacecraft. Each sensor provides range/bearing measurements to a single remote spacecraft throughout the simulation. In order to balance the network and ensure coupling, each spacecraft is sensed by exactly n spacecraft. The weights $W_i = I$ in the virtual center calculations.

***n* > 1 Weighted** Identical to the ***n* > 1 Unweighted** case except $W_i = \mathbf{Y}_i$ when calculating the virtual center.

Switched Unweighted Each spacecraft is equipped with a single AFF sensor and the infinite horizon scheduling algorithm described in Section B is used. The weighting matrix $W_i = I$ when calculating the virtual center. The utility for determining the optimal measurement sequence is obtained by the center state information for the case when $W_i = I$. This yields the utility $\text{trace}(\mathbf{Y}_{ci}) = \text{trace}(\mathbf{Y}_i)$.

Switched Weighted Same as **Switched Unweighted** except the weighting matrix is $W_i = \mathbf{Y}_i$ when calculating the virtual center and the utility function is the center state information $\text{trace}(\mathbf{Y}_{ci})$ is used when calculating the optimal periodic measurement sequence.

LF Simple leader-follower where spacecraft 1 is the leader and spacecraft 2 through 8 are followers.

Figure 7 shows the mean fuel usage and RMS position error for each case. The results show that the switched network exhibits positioning performance superior to the LF network with decreased fuel usage for the unweighted case and increased fuel usage for the weighted case. Also note that there is a trade off between the robustness of the information weighted virtual center versus the performance of the unweighted weighted virtual center. The performance of the weighted case can likely be improved by tuning the q_i parameter either by hand or adaptively as described in Ref. 7.

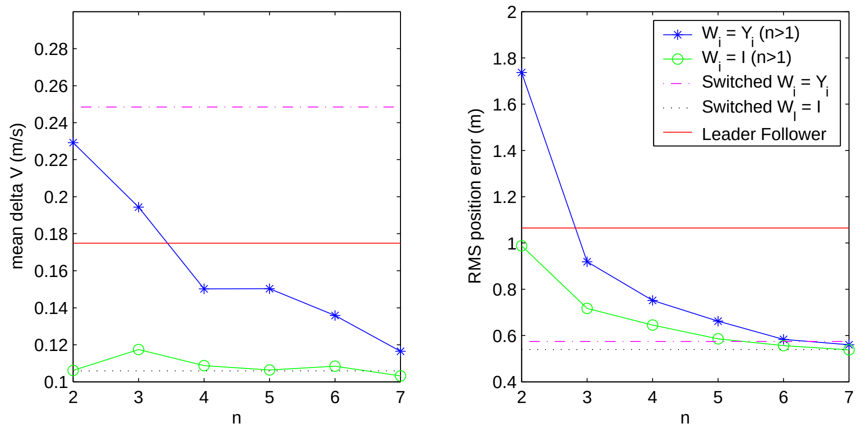

Figure 7. Simulation of distributed unweighted ($W_i = I$) and information weighted ($W_i = \mathbf{Y}_i$) virtual center for spacecraft equipped with $n \in \{2, ..., t\}$ AFF sensors. Traditional leader follower (1 leader, 7 followers) and switched AFF ($n = 1$) also shown.

Simulations have shown that the formation is robust to a fault in a single spacecraft. Here, a fault is considered to be equivalent to a spacecraft performing no correction maneuvers while still transmitting the GPS-type AFF signal such that it is detectable by the rest of the fleet. This situation is equivalent to the faulty spacecraft being designated a leader and the non-faulty spacecraft essentially forms an aggregate follower. Thus, the formation is stable. However, the fault is not detectable under the current architecture. If two or more spacecraft experience a similar fault, the formation will break. In this case, such a fault is detectable by evaluating the center state cost in Eq. 13. If this cost gets large, the formation can be considered to have failed. Although the system is not able to detect which spacecraft have failed, once the cost rises above a certain threshold, the formation can be set to a safe mode and appropriate measures can be taken.

IV. Conclusions and Future Work

The decentralized GNC algorithms originally developed in Ref. 7 have been extended and shown to be robust to a number of faults including: loss of observability/available measurements at the AFF sensor and failure of one or more spacecraft. The sensor scheduling algorithm has been extended to allow for switching constraints dictated by the sensor hardware. Future work includes simulating the system in realtime using the Cornell Formation Flying Testbed and the ObjectAgent middleware. These simulations shall include such faults and realistic constraints on the flight hardware, as well as nonlinear dynamics and disturbances. A communication scheme will also be included to improve the performance of the estimation architecture.

Acknowledgments

Research funded under the NASA Cross-Enterprise Technology Development Program, NASA Grant #NAG5-10440 and the NASA Graduate Student Research Program (JPL).

References

[1] "The Micro-Arcsecond X-ray Imaging Mission," http://maxim.gsfc.nasa.gov.

[2] "The Terrestrial Planet Finder Mission," http://tpf.jpl.nasa.gov.

[3] "The Stellar Imager Mission," http://hires.gsfc.nasa.gov/si/.

[4] "NASA New Millennium Program's Space Technology 9 Project," http://nmp.jpl.nasa.gov/st9/.

[5] Aung, M., Purcell, G., Tien, J., Young, L., Amari, L., Srnvasan, J., Ciminera, M., and Chong, Y., "Autonomous Formation-Flying Sensor for the StarLight Mission," IPN progress report 42-152, JPL, Feb. 2003.

[6] Hadaegh, F., Scharf, D., and Ploen, S., "Initialization of Distributed Spacecraft for Precision Formation Flying," *Proceedings of the IEEE Conference on Control Applications*, Vol. 1, IEEE, June 2003.

[7] McLoughlin, T. H. and Campbell, M. E., "Hybrid Leader Follower and Sensor Scheduling for Large Spacecraft Networks," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA, Wahington, DC, Aug. 2004.

[8] Breger, L., Ferguson, P., How, J., Thomas, S., McLoughlin, T., and Campbell, M., "Distributed Control of Formation Flying Spacecraft Built on OA," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA, Wahington, DC, Aug. 2003.

[9] "ObjectAgent v3.1 User's Guide," Tech. rep., Princeton Satellite Systems, Oct. 2003.

[10] Thurman, R. and Worfolk, P., "The Geometry of Halo Orbits in the Circular Restricted Three-Body Problem," Geometry center research report gcg95, University of Minnesota, 1996.

[11] Richardson, D., "Analytic Construction of Periodic Orbits About the Collinear Points," *Celestial Mechanics*, Vol. 22, 1980, pp. 241–253.

[12] Scharf, D., Hadaegh, F., and Kang, B., "On the Validity of the Double Integrator Approximation in Deep Space Formation Flying," *Int. Symp. Formation Flying Missions & Technologies*, Toulouse, France, 2002.

[13] Tillerson, M., Breger, L., and How, J., "Distributed Coordination and Control of Formation Flying Spacecraft," *Proceedings of the IEEE American Control Conference*, IEEE, June 2003.

[14] Mutambara, A., *Decentralized Estimation and Control for Multisensor Systems*, CRC Press, 1998.

[15] Bittanti, S., Colaneri, P., and Nicolao, G. D., "The Periodic Riccati Equation," *The Riccati Equation*, edited by S. Bittanti, A. Laub, and J. Willems, Springer-Verlag, New York, 1991, pp. 127–162.

[16] Gupta, V., Chung, T., Hassibi, B., and Murray, R., "Sensor Scheduling Algorithms Requiring Limited Computation," *International Conference on Acoustics, Speech, and Signal Processing*, 2004, submitted.