Investigation for Improving Global Positioning System (GPS) Orbits Using a Discrete Sequential Estimator and Stochastic Models Of Selected Physical Processes

Clyde C. Goad and C. David Chadwell
Department of Geodetic Science and Surveying

Goddard Space Flight Center
Greenbelt, Maryland 20771

Grant No. NAG 5-2041
Annual Report

August 1993
Investigation for Improving Global Positioning System (GPS) Orbits Using a Discrete Sequential Estimator and Stochastic Models Of Selected Physical Processes

Clyde C. Goad and C. David Chadwell
Department of Geodetic Science and Surveying

Goddard Space Flight Center
Greenbelt, Maryland 20771

Grant No. NAG 5-2041
Annual Report
RF Project No. 760126/726487

August 1993
Introduction

GEODYNII is a conventional batch least-squares differential corrector computer program with deterministic models of the physical environment. Conventional algorithms have been used to process differenced phase and pseudorange data to determine eight-day GPS orbits with several meter accuracy (Schenewerk, 1990). However, random physical processes drive the errors whose magnitudes prevent improving the GPS orbit accuracy. To improve the orbit accuracy, these random processes should be modeled stochastically. The conventional batch least-squares algorithm cannot accommodate stochastic models, only a stochastic estimation algorithm is suitable, such as a sequential filter/smoother. Also, GEODYNII cannot currently model the correlation among data values. Differenced pseudorange, and especially differenced phase, are precise data types that can be used to improve the GPS orbit precision (Counselman et al., 1989).

To overcome these limitations and improve the accuracy of GPS orbits computed using GEODYNII, we proposed to develop a sequential stochastic filter/smoother processor by using GEODYNII as a type of trajectory preprocessor. Our proposed processor is now completed. It contains a correlated double difference range processing capability, first order Gauss Markov models for the solar radiation pressure scale coefficient and $\gamma$-bias acceleration, and a random walk model for the tropospheric refraction correction.

The development approach has been to interface the standard GEODYNII output files (measurement partials and variationals) with software modules containing the stochastic estimator, the stochastic models, and a double differenced phase range processing routine. Thus, no modifications to the original GEODYNII software have been required. A schematic of the development is shown in Figure 1. The observational data are edited in the preprocessor and the data are passed to GEODYNII as one of its standard data types. A reference orbit is determined using GEODYNII as a batch least-squares processor and the GEODYNII measurement partial (FTN90) and variational (FTN80, V-matrix) files are generated. These two files along with a control statement file and a satellite identification and mass file are passed to the filter/smoother to estimate time-varying parameter states at each epoch, improved satellite initial elements, and improved estimates of constant parameters.
Figure 1. Flowchart showing the procedure as currently developed

Background

The following discussion assumes some familiarity with filtering and smoothing theory as developed, for example in Brown (1983) and Gelb (1974). Additional familiarity is assumed with the square root information filtering and smoothing algorithm as developed in Bierman (1977) and implemented by Swift (1987). Any departures in our implementation from Swift's formulation are
explained here in detail. Some topics are expanded here to clarify and to supplement the discussion in these earlier references. Particular attention has been focused on showing the common foundation of the information and covariance filters. The efficiency of the Householder Transformation to compute an equivalent square upper triangular matrix from a larger rectangular matrix is discussed. Also, the quantities that define the first-order Gauss Markov and random walk models are clearly derived.

The discrete form of the stochastic state equations are:

\[ \Delta x_{j+1} = \Phi_j \Delta x_j + G \omega_j \]  \hspace{1cm} (1)

where

- \( \Delta x_j \) is the state at time \( t_j \)
- \( \Phi_j = \Phi(t_{j+1}, t_j) \) is the nonsingular state transition matrix relating the state at \( t_j \) to the state \( t_{j+1} \).
- \( \omega_j \) is the vector of white noise process terms with a nonsingular covariance matrix \( Q_j \) with \( \text{dim} \omega \leq \text{dim} \Delta x \).
- \( G \) maps the source white noise process into the state with \( \text{dim} \Delta x \).

The discrete form of the linear measurement model is:

\[ z_j = A_j \Delta x_j + v_j \]  \hspace{1cm} (2)

where

- \( z_j \) is the vector of measurements at time \( t_j \)
- \( A_j \) is the matrix of partial derivatives of the measurement model w.r.t. the state at \( t_j \)
- \( v_j \) is the vector of measurement noise with the covariance \( P_0 \).

The observations are decorrelated and whitened so that \( P_0 = I \). This is done without a loss of generality. A set of observations with \( P_0 = I \) can be constructed. The procedure will be given later.

A solution to this problem was first proposed by Kalman in early 1960's (Kalman, 1960; Kalman and Bucy, 1961). A solution for the state and its covariance can be derived by applying Bayes'
rule. This derivation can be found in Maybeck (1979). The results are repeated here with a slight change in notation.

\[ \hat{P}_{j+1} = \left[ \hat{P}_j + A_j^T A_j \right]^{-1} \]  

(3)

\[ \Delta \hat{x}_j = \hat{P}_{j+1} \left[ \hat{P}_j \Delta \hat{x}_j + A_j^T z_j \right] \]  

(4)

The '-' symbol refers to the propagated (predicted) estimate of the state and covariance at \( t_j \). The '+' symbol refers to the estimate of the state or covariance after incorporating the measurement at \( t_j \). The state \( \Delta \hat{x}_j \) is propagated from \( t_j \) to time \( t_{j+1} \), using equation (1). The covariance \( \hat{P}_j \) is propagated to \( t_{j+1} \), by

\[ \hat{P}_{j+1} = \Phi_j \hat{P}_j \Phi_j^T + G_j Q_j G_j^T \]  

(5)

Notice that the inverse of \( \hat{P}_{j+1} \) is required in equations (3) and (4). To avoid the inversion of \( \hat{P}_{j+1} \) at each step, a direct propagation of \( \hat{P}_{j+1} \) is desired. This can be developed by applying the following lemma to equation (5).

\[ \left( A + X^T Y \right)^{-1} = A^{-1} - A^{-1} X^T \left( I + Y A^{-1} X^T \right)^{-1} Y A^{-1} \]  

(6)

where

\[ A = \Phi_j \hat{P}_j \Phi_j^T \]

\[ X^T = G_j Q_j \]

\[ Y = G_j^T \]

and defining

\[ M_{j+1} = \Phi^T (t_{j+1}, t_j) \hat{P}_j^{-1} \Phi(t_{j+1}, t_j) \]  

(7)
\[ \mathbf{P}_{j+1} = \mathbf{M}_{j+1} - \mathbf{C}_j \mathbf{G}_j^T \mathbf{M}_{j+1}, \]  
(8)

where the gain \( \mathbf{C}_j \) is:

\[ \mathbf{C}_j = \mathbf{M}_{j+1} \mathbf{G}_j \left[ \mathbf{G}_j^T \mathbf{M}_{j+1} \mathbf{G}_j + \mathbf{Q}^{-1}_j \right]^{-1} \]  
(9)

To propagate and update the state equations in terms of \( \mathbf{P}_{j} \) and \( \mathbf{P}^{-1}_j \), the state estimates are replaced by

\[ \mathbf{y}_j = \mathbf{P}^{-1}_j \Delta \mathbf{x}_j \]  
(10)
\[ \hat{\mathbf{y}}_j = \hat{\mathbf{P}}^{-1}_j \Delta \mathbf{x}_j \]  
(11)

The state update and propagation equations are:

\[ \hat{\mathbf{y}}_j = \mathbf{y}_j + \mathbf{A}_j^T \mathbf{z}_j \]  
(12)
\[ \mathbf{y}_{j+1} = \left[ \mathbf{I} - \mathbf{C}_j \mathbf{G}_j^T \right] \Phi(t_{j+1}, t_j) \hat{\mathbf{y}}_j \]  
(13)

The state estimates can be found at any time by solving equations (10) and/or (11) for \( \Delta \mathbf{x}_j \) and/or \( \Delta \mathbf{\hat{x}}_j \). Equations (8) to (13) are an algorithm to solve the problem defined by equations (1) and (2). Here, the inverse covariance is propagated. This algorithm is sometimes called a Bayes' filter. The inverse covariance is also called an information matrix leading to the name information filter. This algorithm requires computing the inverse of an \( n \times n \) matrix where \( n \) is the number of states. The state estimate covariance can be completely uncertain since in the inverse of \( \mathbf{P}_0 \) the elements of the matrix become zero. This algorithm is most efficient when the number of measurements, \( m \), is relatively larger than the number of states, \( n \), and when the solutions for the state and covariance are needed infrequently.

The usual Kalman filter can be derived from equations (3) and (4) by applying the matrix lemma:

\[ \left[ \mathbf{P}^{-1}_j + \mathbf{A}_j^T \right]^{-1} = \mathbf{P} - \mathbf{P} \mathbf{A}_j \left[ \mathbf{A}_j \mathbf{P} \mathbf{A}_j^T \right]^{-1} \mathbf{A}_j \]  
(14)
These optimal filters, either the Bayes' or Kalman, exhibit numerical instabilities that cause the state estimates to diverge (Bierman and Thornton, 1977). More numerically stable gain matrix expressions have been derived for both the covariance and information matrix forms (Maybeck ibid.). However, these require a significant number of additional matrix computations and are thus not completely satisfactory. A more comprehensive approach was to reformulate the filter algorithm in terms of square roots of the covariance or the information matrix. The square root filter maintains numerical accuracy to approximately the same number of digits with half the word length required by a conventional non-square root algorithm.

The square root (or more correctly the Cholesky factorization) of an $n \times n$ matrix $N$ is defined as:

$$N = SS^T$$

where $S$ is the Cholesky factor of $N$.

The square roots are not unique. Any orthogonal transformation $(T)$ of a square root matrix $S$ is also a square root of $N$. The useful properties of the orthogonal transformation can be shown by factoring $\hat{P}_j^{-1}$ into the product of its Cholesky factors

$$\hat{P}_j^{-1} = \hat{R}_j^T \hat{R}_j,$$  

and thus, $\hat{P}_j^{-1}$ becomes:

$$\hat{P}_j^{-1} = \hat{R}_j^T \hat{R}_j = \hat{R}_j^T \hat{R}_j + A_j^T A_j$$

where

$$\hat{R}_j = \begin{bmatrix} \hat{R}_j^T \\ A_j \end{bmatrix}$$

Now

$$\begin{bmatrix} \hat{R}_j \\ 0 \end{bmatrix} = T \begin{bmatrix} \hat{R}_j^T \\ A_j \end{bmatrix}$$

The transformation $T$ is an orthogonal transformation. Its columns form an orthonormal basis for $\hat{R}_j$ of $n$ vectors since $\hat{R}_j$ has rank of $n$. The first $n$ vectors span the range space of $\hat{R}_j$ and the vectors $n+1$ to $n+m$ are orthogonal to this spanning set. Thus, the last $m$ rows of $\hat{R}_j$ are zero. Also, the basis vectors were chosen in a manner that $\hat{R}_j$ is upper triangular. A Householder
Transformation T is used to compute $\hat{R}_j$ (Bierman, ibid.). This allows the square root matrices of dimension $(n+m)$ by $n$ to be transformed to an equivalent form of an upper triangular square matrix of dimension $n$.

**Square Root Information Filter and Smoother (SRIF/SRIS)**

In application of the Householder Transformation to an augmented matrix is the essence of the Square Root Information Filter (SRIF). The equations are most easily constructed using Bierman's 'data equation' point of view. The problem is treated as a least-squares problem where the least-squares functional is to minimized. This is accomplished by applying a Householder Transformation assuming the state at $t_j$ to be a priori information and augmented with the measurements at $t_j$. Thus,

$$
\tilde{T}_j\begin{bmatrix}
\hat{R}_j & \hat{z}_j \\
A_j & z_j
\end{bmatrix} = 
\begin{bmatrix}
\hat{R}_j & \hat{z}_j \\
0 & e_j
\end{bmatrix}
$$

where the 'data equations' are defined as

$$
\tilde{z}_j = \hat{R}_j\Delta\tilde{x}_j \quad (21) \\
\hat{z}_j = \hat{R}_j\Delta\hat{x}_j \quad (22)
$$

Swift has shown the equivalence of equation (20) with the more conventional formulation of equations (3) and (4).

The propagation of the state and the covariance were given in equations (1) and (5). These can also be incorporated into the SRIF by defining 'data equations' and applying the Householder Transformations. The details of the derivations can be found in Bierman (ibid.). The results are repeated here.

$$
\tilde{T}_{j+1}\begin{bmatrix}
\hat{R}_\omega(j) & 0 & \hat{z}_\omega(j) \\
-\hat{R}_j\Phi_j^{-1}G & \hat{R}_j\Phi_j^{-1} & \tilde{z}_{j+1}
\end{bmatrix} = 
\begin{bmatrix}
\hat{R}_\omega(j+1) & \hat{R}_\omega(j+1) & \hat{z}_\omega(j+1) \\
0 & \hat{R}_{j+1} & \tilde{z}_{j+1}
\end{bmatrix}
$$

where the 'data equation' for the noise term $\omega$ is...
\[ z_\omega(j) = R_\omega(j)\omega(j) \] (24)

Swift has also shown the equivalence of equation (23) to equations (1) and (5).

Bierman further partitions the propagation equation (1) into stochastic states, dynamic states and bias states. Equation (1) can be written as:

\[
\begin{bmatrix}
\Delta p \\
\Delta x \\
\Delta y_{j+1}
\end{bmatrix} =
\begin{bmatrix}
M & 0 & 0 \\
V_p & V_x & V_y \\
0 & 0 & I
\end{bmatrix}
\begin{bmatrix}
\Delta p \\
\Delta x \\
\Delta y_j
\end{bmatrix} +
\begin{bmatrix}
\omega_j \\
0 \\
0
\end{bmatrix}
\] (25)

where
- \(\Delta p\) is the correlated process noise states
- \(\Delta x\) states that vary with time by not explicitly influenced by process noise.
- \(\Delta y\) bias (constant) parameters.
- \(V_p, V_x, V_y\), are transition matrix elements.

The dynamic parameters can be redefined in the form of pseudo epoch state parameters. This dynamic model definition allows the variationals and measurement partials from a batch differential corrector orbit determination program (e.g., GEODYNII) to be used directly in the filter algorithms. The state equations now become

\[
\begin{bmatrix}
\Delta p \\
\Delta x \\
\Delta y_{j+1}
\end{bmatrix} =
\begin{bmatrix}
M & 0 & 0 \\
V_p & I & 0 \\
0 & 0 & I
\end{bmatrix}
\begin{bmatrix}
\Delta p \\
\Delta x \\
\Delta y_j
\end{bmatrix} +
\begin{bmatrix}
\omega_j \\
0 \\
0
\end{bmatrix}
\] (26)

where
- \(V_p = V_p(t_{j+1}, t_j) = V_x^{-1}(t_{j+1}, T_0)V_p(t_{j+1}, t_j)\)
- \(V_x^{-1}(t_{j+1}, T_0)\) is the inverse of the state transition matrix interpolated from the GEODYNII V-matrix file (FTN80)
- \(V_p(t_{j+1}, t_j)\) is the transition matrix of the time-varying parameters from \(t_j\) to \(t_{j+1}\).
The transformation in equation (20) is written as a two step transformation that saves storing a block of \( nx \times (np+nx) \) zeroes that would be present if equation (20) was used in its original form.

\[
\begin{bmatrix}
\tilde{R}_p & \tilde{R}_{px} & \tilde{R}_{py} & \tilde{z}_p \\
T_{px} & \tilde{R}_{xp} & \tilde{R}_{xy} & \tilde{z}_x \\
A_p & A_x & A_y & z
\end{bmatrix} = \begin{bmatrix}
\hat{R}_p & \hat{R}_{px} & \hat{R}_{py} & \hat{z}_p \\
0 & \hat{R}_x & \hat{R}_{xy} & \hat{z}_x \\
0 & 0 & \hat{A}_y & \hat{z}
\end{bmatrix}
\] (27)

\[
\begin{bmatrix}
\tilde{R}_y & \tilde{z}_y \\
\tilde{A}_y & \tilde{z}
\end{bmatrix} = \begin{bmatrix}
\hat{R}_y & \hat{z}_y \\
0 & e
\end{bmatrix}
\] (28)

The mapping equation (23) becomes, assuming \( z_w \) is zero

\[
\begin{bmatrix}
-R_w M & R_w & 0 & 0 & 0 \\
\hat{T}_p & -\hat{R}_{px} V_p & 0 & \hat{R}_{py} & \hat{z}_p \\
-\hat{R}_x V_p & 0 & \hat{R}_x & \hat{R}_{xy} & \hat{z}_x
\end{bmatrix} = \begin{bmatrix}
\hat{R}_p & \hat{R}_{pp} & \hat{R}_{px} & \hat{R}_{py} & \hat{z}_p \\
0 & \hat{R}_p & \hat{R}_{px} & \hat{R}_{py} & \hat{z}_p \\
0 & \hat{R}_{sp} & \hat{R}_x & \hat{R}_{xy} & \hat{z}_x
\end{bmatrix} (29)
\]

The subroutine in Bierman (ibid., 155-157) neglected the upper triangular elements of \( \hat{R}_p \) above the diagonal. A subroutine to compute right side of equation (29) including the neglected off-diagonal elements of \( \hat{R}_p \) is given in Appendix B.

The smoothing process is a backward filter of the forward filter results. For the orbit determination problem, the fixed interval smoother is appropriate. For Kalman filtering, the Rauch-Tung-Streibel (RTS) smoothing algorithm is widely used (Brown, ibid.). A general formulation for inverse covariance smoothing is given by Maybeck (ibid.). The Square Root Information Smoother (SRIS) is given by Bierman (ibid.). The equation for the implementation of Bierman's pseudo epoch formulation is given by Swift (ibid.). Swift also shows the equivalence of the SRIS to the RTS smoother. The SRIS equation is

\[
T_{px}^{f^*} \begin{bmatrix}
\hat{R}_p & \hat{R}_p^* + \hat{R}_p^* M + \hat{R}_p^* V_p & R_p^* R_p^* V_p & \hat{R}_p^* R_p^* \hat{z}_p \\
0 & \hat{R}_p^* M + \hat{R}_p^* V_p & R_p^* R_p^* \hat{z}_p \\
0 & 0 & \hat{R}_p^* R_p^* \hat{z}_p
\end{bmatrix}_{j+1} = \begin{bmatrix}
\hat{R}_p & \hat{R}_p & \hat{R}_p & \hat{R}_p & \hat{z}_p \\
0 & \hat{R}_p & \hat{R}_p & \hat{R}_p & \hat{z}_p \\
0 & 0 & \hat{R}_p & \hat{R}_p & \hat{z}_p
\end{bmatrix}_{j+1}
\] (30)
The top row of the matrix on the right side is not needed again, but the other terms are combined with the smoothing coefficients at $t_j$ to smooth back to $t_{j-1}$.

From the general expression of the data equation, and the relationship of the covariance to the square roots of the information matrix, the solution for the states and their covariances for either the filter or smoothing operations are

$$x_j = R_j^{-1}z_j$$

$$P_j = R_j^{-1}R_j^{-T}$$

For filtering, the right sides of equations (27) and (28) are solved for the unknowns and the covariances as

$$\begin{bmatrix} \Delta p \\ \Delta x \\ \Delta y_j \end{bmatrix} = \begin{bmatrix} R_x & R_{px} \\ 0 & R_x \end{bmatrix}^{-1} \begin{bmatrix} (z_p) - (R_y)R_y^{-1}z_y \\ z_x \\ R_y^{-1}z_y \end{bmatrix}$$

$$R_j^{-1} = \begin{bmatrix} R_p & R_{px} \\ 0 & R_x \end{bmatrix}^{-1} - \begin{bmatrix} R_p & R_{px} \\ 0 & R_x \end{bmatrix}^{-1} \begin{bmatrix} R_{px} \\ R_{px} \\ R_{xy} \end{bmatrix} R_y^{-1} \begin{bmatrix} R_p & R_{px} \\ 0 & R_x \end{bmatrix}$$

$$P_{px} = \begin{bmatrix} R_p & R_{px} \\ 0 & R_x \end{bmatrix}^{-1} \begin{bmatrix} R_p & R_{px} \\ 0 & R_x \end{bmatrix}^{-T} + \begin{bmatrix} R_p & R_{px} \\ 0 & R_x \end{bmatrix}^{-1} \begin{bmatrix} R_{py} \\ R_{xy} \end{bmatrix} R_y^{-1} \begin{bmatrix} R_p & R_{px} \\ 0 & R_x \end{bmatrix}$$

$$P_y = R_y^{-1}R_y^{-T}$$

For the smoothing problem, the right side of (30) is used in equations (32)-(36) with the smoothed value of $z_y$ and $R_y$. The smoothed values of $z_y$ and $R_y$ are the values at the last filter step.

**Time-Varying Stochastic Parameter Models**

Many physical processes can be modeled using one of the Gauss Markov filters easily constructed by filtering white noise through a simple filter. The processes that are of interest here are the...
first-order Gauss Markov model and the random walk model. The discrete mathematical expression for these models is now derived from their continuous forms. The first-order Gauss Markov process describes the physical process where the state at $t_{j+1}$ depends only on the previous state at $t_j$. This process can be described by the stochastic differential equation.

\[
\frac{dp(t)}{dt} - \frac{1}{\tau} p(t) = \omega \tag{37}
\]

where

- $\tau$ is the correlation time
- $\omega$ is the white Gaussian noise with zero mean and covariance $Q$,

\[
E(\omega_j) = 0, \quad E(\omega_j, \omega_k) = Q \delta(j - k) = q_{con} \delta(j - k) \quad \text{where } q_{con} \text{ is the continuous spectral density.}
\]

The variance of $p$, $\sigma_p^2$ satisfies the differential equation

\[
\frac{d\sigma_p^2(t)}{dt} = -\frac{2}{\tau} \sigma_p^2(t) + q_{con} \tag{38}
\]

If the process is allowed to continue for a time interval several multiples longer than $\tau$, the $E(p^2(t))$ will approach a limit and $\frac{d\sigma_p^2(t)}{dt}$ will approach zero. This is the steady state variance.

By setting $\frac{d\sigma_p^2(t)}{dt} = 0$ and solving equation (38) for steady state $\sigma_p^2(t_{st})$ is found

\[
\sigma_p^2(t_{st}) = \frac{\tau}{2} q_{con} \tag{39}
\]

From state space methods the solution of equation (37) is

\[
p(j + 1) = M(j + 1, j)p(j) + \int_j^{j+1} M(j, \lambda) \omega(\lambda) d\lambda \tag{40}
\]

and covariance of $p$ is

\[
C_p = MP_pM^T + \int_j^{j+1} M(j + 1, \lambda)Q(\lambda)M^T(j + 1, \lambda) d\lambda \tag{41}
\]
The matrix $M$ is the state transition matrix which must satisfy the relationships

$$M = \frac{1}{\tau} M \quad \text{and} \quad M(j,j) = 1$$

(42)

The solution for $M$ is

$$M = e^{\frac{\Delta t}{\tau}}$$

(43)

where $\Delta t = t_{j+1} - t_j$

Thus the discrete form of the state update becomes

$$\Delta p(j + 1) = M(j + 1,j)\Delta p(j) + \omega_j$$

(44)

This is the top row of equation (26).

Now the solution for the discrete covariance update is derived from

$$q_{dix} = \int_{t_j}^{t_{j+1}} q_{con} e^{-\frac{2(x-y)}{x}} d\lambda = -q_{con} \left[ e^{-\frac{2}{\tau}t_j} \right]^{t_{j+1}} = q_{con} \frac{\tau}{2} \left( 1 - e^{-\frac{2\Delta t}{\tau}} \right)$$

(45)

The random walk model is a special case of the first order Gauss Markov where $\tau \to \infty$. The discrete state update becomes

$$\Delta p(j + 1) = \Delta p(j) + \omega_j$$

(46)

The discrete covariance is found from

$$q_{dix} = \lim_{\tau \to -} q_{con} \frac{\tau}{2} \left( 1 - e^{-\frac{2\Delta t}{\tau}} \right) = q_{con} \lim_{\tau \to -} \left( 1 - \frac{2\Delta t}{\tau} + \left( \frac{2\Delta t}{\tau} \right)^2 \right)$$
\[
= q_{\text{con}} \lim_{\tau \to \infty} \left( \Delta t + \sum_{i=1}^{\infty} \text{constant}_i \left( \frac{1}{\tau} \right)^i \right) = q_{\text{con}} \Delta t
\] 

(47)

So, to implement the first-order Gauss Markov process, the correlation time \( \tau \) and the continuous process noise variance \( q_{\text{con}} \) must be specified. The matrix \( M \) is computed using equation (43) and the \( q_{\text{dis}} \) from equation (45). The random walk model is specified by defining the continuous process noise variance \( q_{\text{con}} \), here \( M = I \), and \( q_{\text{dis}} \) is computed using equation (47).

**Solar Radiation Pressure Scale Coefficient and Y-bias Acceleration Model**

The orbit related stochastic parameters modeled in OSUORBFS are first-order Gauss Markov models for the solar radiation pressure scale coefficient and the \( Y \)-bias acceleration. The \( V_p \) matrix that maps the effects of the stochastic parameters \( \Delta p_j \), on the epoch state parameters \( \Delta x_j \), is derived following the scheme of Swift (ibid.). The \( V_p \) matrix has dimension \( nx \) (number of pseudo epoch state parameters) by \( nd \) (number of orbit-related stochastic parameters) and has the general form of

\[
V_p(t_{j+1},t_j) = \\
\begin{bmatrix}
\Phi_{K_r} & 0 & \cdots & \cdots & 0 \\
0 & \ddots & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & 0 & \ddots \\
0 & \cdots & \cdots & 0 & \Phi_{Y-bias} \\
\end{bmatrix}
\]

(48)

The \( i \)th satellite contribution \( \Phi_{K_r}^i = \Phi_{K_r}^i (t_{j+1},t_j) \), a 4 by 2 matrix, to \( V_p \) is computed as

\[
\Phi_{K_r}^i (t_{j+1},t_j) = \Phi^{-1}_e (t_{j+1},t_j) \begin{bmatrix}
\frac{\partial \tau(t_{j+1})}{\partial K_{\eta}(t_j)} & \frac{\partial \tau(t_{j+1})}{\partial K_{\eta}(t_j)} \\
\frac{\partial \eta(t_{j+1})}{\partial K_{\eta}(t_j)} & \frac{\partial \eta(t_{j+1})}{\partial K_{\eta}(t_j)} \\
\frac{\partial \eta(t_{j+1})}{\partial K_{\eta}(t_j)} & \frac{\partial \eta(t_{j+1})}{\partial K_{\eta}(t_j)} \\
\frac{\partial \eta(t_{j+1})}{\partial K_{\eta}(t_j)} & \frac{\partial \eta(t_{j+1})}{\partial K_{\eta}(t_j)} \\
\end{bmatrix}
\]

(49)
where

$$\Phi^{-1}(t_{j+1}, t_j)$$ is the state transition matrix interpolated from the GEODYNII output V-matrix file (FTN80) variationals. See also equation (26).

The partial derivatives of the position and velocity at $t_{j+1}$ with respect to the solar radiation pressure scale coefficient and the $y$-bias acceleration at $t_j$ are approximated using a second-order Taylor Series expansion.

$$\frac{\partial \mathbf{r}(t_{j+1})}{\partial \mathbf{K}_a(t_j)} = \frac{\Delta t^2}{2} \frac{\partial \mathbf{r}(t_j)}{\partial \mathbf{K}_a(t_j)}$$

$$\frac{\partial \mathbf{v}(t_{j+1})}{\partial \mathbf{K}_a(t_j)} = \Delta t \frac{\partial \mathbf{v}(t_j)}{\partial \mathbf{K}_a(t_j)} - \frac{\Delta t^2}{2} \frac{\partial \mathbf{v}(t_j)}{\partial \mathbf{K}_a(t_j)} \mathbf{B}_{k_a} l = 1, 2$$

(50)

where

$$\mathbf{B}_{k_a} = \left[ \begin{array}{cc} 1 & 0 \\ -\frac{1}{\tau_{k_a}} & 0 \\ 0 & -\frac{1}{\tau_{k_a}} \end{array} \right]$$

$$\mathbf{B}_{k_a} = 0$$ for a random walk.

Since the angle between the $x$ and $y$ satellite axes is not easily estimated it is not modeled in OSUORBFS. A 90 degree angle is assumed. Thus the following differs slightly from Swift.

$$\begin{bmatrix} \frac{\partial \mathbf{r}(t_j)}{\partial \mathbf{K}_a(t_j)} & \frac{\partial \mathbf{v}(t_j)}{\partial \mathbf{K}_a(t_j)} \end{bmatrix} = R_a \begin{bmatrix} a^m_{shape} & 0 \\ 0 & shape \\ a^m_{shape} & 0 \end{bmatrix}$$

(51)

where

$a^m_x$ and $a^m_z$ are the accelerations along the satellite $x$ and $z$ axis respectively.

The ROCK4 and ROCK42 models are used to compute $a^m_x$ and $a^m_z$.

$shape$ is either 0 or 1 depending if the sun is obstructed by the earth from view of
the satellite.

\( \mathbf{R}_s \) is a matrix transformation from the satellite axis system to the True of Reference Date (TORD) inertial Cartesian reference system.

The ROCK4 and ROCK42 models (Fliegel and Gallini, 1992), the matrix transformation \( \mathbf{R}_s \), and the computation of shape require the sun-earth-satellite positions. The mean of date (MOD) positions of the sun and earth were computed using the closed form expressions of Fliegel and Harrington (1993).

**Tropospheric Refraction Correction**

The measurement-related stochastic parameter modeled in OSUORBFS is a random walk model for the refraction correction (Tralli et al., 1988; Herring et al., 1990). This model is defined by equations (46) and (47).

**Double Difference Observable Decorrelation and Whitening**

The full covariance matrix for the double difference range data is constructed. The observations are then decorrelated and whitened. The general form of the observation equations as defined in equation (2) is

\[
\mathbf{z} = \mathbf{A} \mathbf{x} + \mathbf{v} \quad (52)
\]

Here, the observation error \( \mathbf{v} \) has a zero mean, \( E(\mathbf{v}) = 0 \), but is correlated, \( E(\mathbf{vv}^T) = \mathbf{P}_v \).

A set of uncorrelated observations with unit covariance can be constructed from the lower triangular square root of \( \mathbf{P}_v \).

\[
\mathbf{P}_v = \mathbf{L}_v \mathbf{L}_v^T \quad (53)
\]

Here, \( \mathbf{L}_v \) can be computed by a lower Cholesky factorization of \( \mathbf{P}_v \). The desired independent set of observations is

\[
\mathbf{L}_v^{-1} \mathbf{z} = \mathbf{L}_v^{-1} \mathbf{A} \mathbf{x} + \mathbf{L}_v^{-1} \mathbf{v} \quad (54)
\]
At any particular epoch, the \( m = (\text{#stations}-1) \times (\text{#satellites}-1) \) linear independent double difference range data types can be formed. These \( m \) observations are independent in the sense of linear algebra, but are statistically correlated. Each of the \( m \) observations has the form (GEODYNII measurement type=87)

\[
[(s_1 \rightarrow t_1) - (s_2 \rightarrow t_1)] - [(s_1 \rightarrow t_2) - (s_2 \rightarrow t_2)]
\]

(55)

where

\((s_1 \rightarrow t_1)\) etc., are the satellite-station range observations (e.g., satellite 1 to station 1).

For small regional networks, a single satellite station pair is selected as the base satellite-station and the \( m \) observations are constructed by differencing the remaining satellite-stations with the base pair. For a global network, the distance between stations may prevent using a single base pair to construct all observations at that epoch. Thus, no consistent numerical structure exists that would permit a symbolic construction of the decorrelated measurement set. \( P_v \) and \( L_v^{-1} \) must be computed numerically at each epoch. \( P_v \) is computed using conventional error propagation

\[
P_v = \sigma_r^2 G G^T
\]

(56)

where

\( \sigma_r^2 \) is the standard deviation of the single one-way range measurement

\( G \) is the matrix of partial derivatives of the observation equation with respect to the one-way range. This matrix contains elements of -1,0,1 which are the linear combination of one-way ranges that define the double difference

The decorrelated observation set with unit covariance is obtain from equations (53) and (54).

**OSUORBFS**

The program OSUORBFS is designed to filter and smooth the GEODYNII batch solutions. OSUORBFS requires from GEODYNII the measurement partials file (FTN90) and the variationals V-matrix file (FTN80). The user must supply a user input file (FSN05) and a file of satellite identification numbers and masses (SATMAS.TAB). The GEODYNII processing proceeds in the usual way with TDF, G2S, and G2E program executions. On the last iteration an
output of the setup deck (FTN05) which contains the current parameter estimates is requested using PUNCH to output the new setup deck in file FTN07. Then FTN07 is modified to include global cards PARFIL and EMATRX to output files FTN90 and FTN80. The maximum iteration numbers for the global (outer) and arc (inner) are set to one on the ENDGLB and REFSYS statements.

Now TDF, G2S, and G2E are executed and FTN06, FTN80, and FTN90 files are output. An alternative approach avoiding the restart of GEODYNII is to force an additional iteration in the first GEODYN execution by increasing the outer/inner iteration maximum counts and decreasing the RMS tolerance. This approach would be less cumbersome to implemented. Additional operational experience is needed to determine which approach is satisfactory.

The user must construct the control file, FSN05, for OSUORBFS. This file contains six control statements (REFSYS, DECORR, FILSMT, UPDTRJ, CONPRT, SATMAS) to control the configuration of the filter/smoother solutions. The six statements are mandatory. These statements are explained in Appendix A.

FSN05 must also contain the parameter labels from FTN06. The measurement partials and variationals form GEODYNII are identified by internal parameter labels as described in the GEODYNII manual volume 5. For OSUORBFS to recognize these partials, the parameter labels as they appear in FTN80 and FTN90 must be specified in FSN05. These labels can be accessed by printing the EMATRIX header record in FTN06 during the last iteration of the GEODYN run. They must be manually edited and placed in FSN05.

FSN05 must also contain the parameter types as defined in the following table.

<table>
<thead>
<tr>
<th>Type #</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>orbit-related stochastic</td>
<td>solar radiation pressure (1st Gauss Markov)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>y-bias acceleration (1st Gauss Markov)</td>
</tr>
<tr>
<td>2</td>
<td>measurement-related stochastic</td>
<td>tropospheric refraction correction (random walk)</td>
</tr>
<tr>
<td>3</td>
<td>pseudoepoch state</td>
<td>satellite initial elements</td>
</tr>
<tr>
<td>4</td>
<td>measurement-related constant</td>
<td>double difference bias, tropospheric refraction correction</td>
</tr>
<tr>
<td>5</td>
<td>orbit-related constant</td>
<td>solar radiation pressure coefficient, y-bias</td>
</tr>
</tbody>
</table>
For each parameter type the apriori standard deviation must be specified. Additionally, for the first-order Gauss Markov model, the continuous process noise standard deviation (\(\sqrt{q_{con}}\)) and the correlation (\(\tau\)) time must be specified. For the random walk model, the continuous process noise standard deviation (\(\sqrt{q_{con}}\)) and a negative correlation time (\(-\tau\), which acts as a flag) must be specified. These are read in a free format.

The order of the parameter types in FSN05 is arbitrary; the file is sorted and the time-varying stochastic parameters are moved to the top of the file to accommodate the space saving implementation of the \(V_p\) as an \(nx \times nd\) matrix.

The stochastic parameters (types 1, 2) are assumed to be zero mean processes. Typically the physical process modeled is not zero mean. The non-zero mean is estimated as a constant (types 4, 5). Thus, the constant (types 4, 5) and the stochastic parameters (types 1, 2) are estimated together. The constant can be estimated without a stochastic parameter, but a stochastic parameter must be estimated with a constant unless of course the process has a zero mean.

Since GEODYNII does not have time-varying models the physical processes are modeled by estimating constants over consecutive segments of time. For example, the tropospheric scale correction in GEODYNII may be modeled over a 24 hour period by estimating a constant over the first 12 hours and another constant over the second 12 hours. In OSUORBFS, the one constant and a time-varying stochastic parameter would be estimated over the entire span of 24 hours. This requires the measurement partials from the two consecutive constant estimates in GEODYNII to be concatenated. This is controlled by the CONPRT statement.

OSUORBFS can be implemented as a conventional least-squares sequential estimator by specifying all parameters as pseudo epoch state (Type 3) and constant parameters (Types 4, 5). The partial file should not be concatenated. The full covariance matrix may be generated. If the GEODYNII solution is to be repeated using OSUORBFS in a sequential least-squares step, then the full covariance matrix should not be formed.
REFERENCES


APPENDIX A
### REFSYS

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td>REFSYS - Specifies the True of Reference Date (TORD) reference system used by GEODYNII. The time <strong>MUST</strong> be the same as the time used on the REFSYS statement in FTN05 (G2S). Only TORD is valid. Mean of Date (MOD) J2000 is not currently implemented</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-26</td>
<td>I6</td>
<td>Year, month, day of reference date (YYMMDD)</td>
<td></td>
</tr>
<tr>
<td>27-30</td>
<td>I4</td>
<td>Hour, minute of reference date (HHMM)</td>
<td></td>
</tr>
<tr>
<td>31-40</td>
<td>D10.8</td>
<td>Seconds of reference date (SS.ssssss)</td>
<td></td>
</tr>
</tbody>
</table>
### DECORR

---

DECORR 1  0.10

---

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td>DECORR - controls the computation of the full covariance matrix and decorrelation for double difference ranges</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>I1</td>
<td>= 0, measurements assumed uncorrelated, measurements whitened by dividing by the GEODYNII supplied weight. = 1, full covariance matrix computed for double differenced ranges, then decorrelated and whitened.</td>
<td></td>
</tr>
<tr>
<td>9-10</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-20</td>
<td>D10.5</td>
<td>standard deviation for a one-way range meters measurement</td>
<td></td>
</tr>
</tbody>
</table>

---
**FILSMT**

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td><strong>FILSMT</strong> - controls the filter/smoothe operations, controls the computation of the estimates/covariances of the stochastic and pseudo epoch parameters (px) and the constants (y)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 8       | II     | = 0, do not filter data  
|         |        | = 1, filter data         |
| 9       | II     | = 0, do not compute the estimate px at each filter step  
|         |        | = 1, compute the estimate px at each filter step         |
| 10      | II     | = 0, do not compute the covariance px at each filter step  
|         |        | = 1, compute the covariance px at each filter step         |
| 11      | II     | = 0, do not compute the estimate y at each filter step  
|         |        | = 1, compute the estimate y at each filter step         |
| 12      | II     | = 0, do not compute the covariance y at each filter step  
|         |        | = 1, compute the covariance y at each filter step         |
| 13      | II     | = 0, do not smooth data  
|         |        | = 1, smooth data         |
| 14      | II     | = 0, do not compute the estimate px at each smoother step  
|         |        | = 1, compute the estimate px at each smoother step         |
15 I1 = 0, do not compute the covariance px at each smoother step
     = 1, compute the covariance px at each smoother step

16 I1 = 0, do not compute the estimate y at the last filter step/first smoother step
     = 1, compute the estimate y at the last filter step/first smoother step

17 I1 = 0, do not compute the covariance y at the last filter step/first smoother step
     = 1, compute the covariance y at the last filter step/first smoother step
**UPDTRJ**

---+---+---+---+---+---+---

**UPDTRJ 1**

---+---+---+---+---+---+---

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td>UPDTRJ - controls the satellite trajectory computation and output in a TORD system to file</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>I1</td>
<td>= 0, trajectory is not updated</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 1, trajectory is updated</td>
<td></td>
</tr>
</tbody>
</table>
### CONPRT

---+---1---+---2---+---3---+---4---+---5---+---6---

**CONPRT 1**

---+---1---+---2---+---3---+---4---+---5---+---6---

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td>CONPRT - controls the concatenation of the piece-wise measurement partials to the first partial location</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>I1</td>
<td>= 0, no concatenation = 1, concatenate</td>
<td></td>
</tr>
</tbody>
</table>
**SATMAS**

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td>SATMAS - controls which satellite number system to use</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>blank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>I1</td>
<td>= 0, GEODYN international satellite id</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 1, OSU modified international satellite id</td>
<td></td>
</tr>
</tbody>
</table>

SATMAS 0

---+---+---+---+---+---+---
SATMAS 0
---+---+---+---+---+---+---
**subroutine hsttp(dt,np,nx,npx,nd,ntot,tau,vp,rw,s,rpsm,v,dm,maxnd,**
.                              maxnx,maxnp,maxobs,maxntr,maxntc,pnstdv)**

Apply a sequence of elementary Householder transformations to
partially triangularize the a posteriori information array,
this process propagates the SRIF from time $t$ to time $t+dt$.

Adapted from 'Factorization Methods for Discrete Sequential

**Version** 9305.1  **Comments**  
**Pgmr.** Modified to compute correctly the $R_p$ matrix. The off-diagonal elements
were missing from original version.

**Variable**  
**Type**  
**i/o**  
**Description**

dt  r*8  i  
Number of stochastic time-varying  
parameters
np  i*4  i  
Number of bias (constant) parameters
nx  i*4  i  
Number of orbit-related stochastic  
parameters
tau(np)  r*8  i  
The first Nd columns of the Vp matrix
 correspond to the dynamic parameters;  
the last Np-Nd columns are omitted
because they are in theory zero.
Vp(nx,nd)  r*8  i  
Process noise standard deviation  
reciprocals
S(nx+2*np,ntot)  r*8  i  
The top Np+Nx rows of S contain the
SRIF array corresponding to the p
and x variables; the bottom p rows
are used to store smoothing related
terms.
Rw(np)  r*8  i  
Upper triangular matrix contains
smoothing related terms from the
$t$-dt to $t$.
Rpsm(np*(np+1)/2)  r*8  i  
Upper triangular matrix contains
smoothing related terms from $t$ to
$t+dt$.

**S on input:**

**S on output:**

<table>
<thead>
<tr>
<th>Np</th>
<th>Np</th>
<th>Np</th>
<th>Np</th>
<th>Np</th>
<th>Np</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tp</th>
<th>Tp</th>
<th>Tp</th>
<th>Tp</th>
<th>Tp</th>
<th>Tp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zw</th>
<th>Zw</th>
<th>Zw</th>
<th>Zw</th>
<th>Zw</th>
<th>Zw</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**implicit none**

**integer*4**

i,j,k,l,np,nd,nx,npx,ntot,j1,j2

**maxnd,ict,index,maxnx,idiag,idiag2**

**integer*4**

maxnp,maxobs,maxntr,maxntc

31
double precision  vp(maxnx,maxnd)
double precision  s(maxntr,maxntc)
double precision  v( maxnp+maxnx+maxobs )
double precision  rw(maxnp),tau(maxnp),dm(maxnp),z,sigma
double precision  alpha, delta, dt, tmp
double precision  rpsm( maxnp*(maxnp+1)/2 )
double precision  pnstdv( maxnp )

z=0.d0
do j=1,np
    if(tau(j).gt.0.d0)then
        !Rw for 1st order Gauss Markov Model
        dm(j)=dexp( -dt/tau(j) )
        rw(j)=1.d0/( pnstdv(j)*dsqrt(1.d0-dm(j)*dm(j)) )
    else
        !Rw for random walk model
        dm(j)=1.d0
        !If tau.lt.0 flag for random walk
        rw(j)=dsqrt( 1.d0/dt )/pnstdv(j)
    endif
endo

idiag=0
idiag2=0
do jl=1,np
    idiag2=idiag2+jl
    if(jl.le.nd)then
        do i=1,npx
            do k=1,nx
                s(i,l)=s(i,l)-s(i,np+k)*vp(k,jl)
            enddo
        enddo
        if(jl.gt.l)then
            ict=npx+jl-I
            idiag=idiag+jl
            index=idiag
            do i=jl,np
                do k=1,nx
                    rpsm(index)=rpsm(index)-s(ict,np+k)*vp(k,jl)
                enddo
                index=index+i
            enddo
        endif
    endif
    idiag=idiag+jl
endo

alpha = -rw(jl)*dm(jl)  !! Assumes an uncorrelated process noise cov.
sigma = alpha*alpha
do i=1,npx
    v(i)=s(i,l)
    sigma = sigma + v(i)*v(i)
endo
sigma = dsqrt( sigma )
alpha = alpha - sigma
ict=jl-1
index=idiag2
rpsm(index) = sigma
sigma = 1.d0/(sigma*alpha)
do j2=2,ntot
    delta=z
    if(j2.eq.ntot)delta=alpha*zw(jl)  !! Assume zero mean
    do i=1,npx
        delta = delta + s(i,j2)*v(i)
    enddo
    delta=delta*sigma
    l=j2-1

32
tmp=delta*alpha
if(j2.gt.np)then
   l=j2
else
   if( j2.le.(np+1-jl) )then
      ict=ict+1
      index=index+ict
      rpsm( index ) = tmp
   endif
endif
s(npx+j1,l)=tmp
do i=1,npx
   s(i,l)=s(i,j2)+delta*v(i)
endo
c s(npx+j1),ntot)=s(npx+j1,ntot)+delta*zw(jl) !! Assume zero
!! mean
delta=alpha*rw(jl)*sigma
s(npx+j1,j1)=rw(j1)+delta*alpha
do i=1,npx
   s(i,np)=delta*v(i)
endo
return
end