ARTICLE NO. 78-1428R

G-80-057

Sequential Estimation Algorithm Using a Continuous UDU^T Covariance Factorization

30007
70023
70030

B.D. Tapley* and J.G. Peters†
The University of Texas, Austin, Texas

A method for propagating the square root of the state-error covariance matrix in lower triangular UDU^T form is described. The propagation method can be combined with the UDU^T measurement incorporation algorithm to obtain a complete square root free triangular estimation algorithm. The method is compared with 1) the UDU^T state transition matrix propagation algorithm; and 2) the conventional sequential estimation algorithms on the basis of estimation accuracy, computational efficiency, and storage requirements by a simulation of LANDSAT-D processing data from the Phase I Global Positioning System. The numerical results indicate that, while slower than the conventional methods, the method proposed here is more efficient than the previous UDU^T factorizations with regard to computer storage and computation time, and leads to the most accurate estimate of any of the methods considered.

Nomenclature

a = semimajor axis of satellite orbit
$\bar{a}_d$ = atmospheric drag acceleration
b = filter model clock bias
b_l = satellite clock bias
b_s = GPS clock bias
c = speed of light
d = ballistic coefficient
e = eccentricity
f = true anomaly
$\bar{g}$ = gravitational acceleration
h = satellite altitude
h_0 = scale height for density model
I = inclination
k = density model scaling factor
n = filter model clock drift
n_l = satellite clock drift
n_s = GPS clock drift
$\bar{r}$ = satellite inertial position vector
t = true time
T = satellite clock indicated time
$\bar{v}$ = satellite inertial velocity vector
$\bar{v}_{rel}$ = velocity vector relative to the atmosphere
v_{rel} = magnitude of $\bar{v}_{rel}$
Y_ρ = range measurement
$Y_{\dot\rho}$ = range-rate measurement
β = correlation parameter for clock model
γ = atmospheric density
γ_0 = atmospheric density at reference altitude
ρ = geometric range
$\dot\rho$ = geometric range-rate
ω = argument of pericenter
Ω = longitude of ascending node

Presented as Paper 78-1428 at the AIAA/AAS Astrodynamics Conference, Palo Alto, Calif., Aug. 7-9, 1978; submitted Sept. 15, 1978; revision received Aug. 14, 1979.

Index categories: Spacecraft Navigation, Guidance and Flight-Path Control; Computer Communications, Information Processing and Software; Spacecraft Systems.

*Professor, Dept. of Aerospace Engineering & Engineering Mechanics. Associate Fellow AIAA.

†Research Assistant, Dept. of Aerospace Engineering and Engineering Mechanics. Student Member AIAA.

Introduction

SQUARE root filter formulations have been proposed as a means of eliminating the problem of filter divergence in the real-time application of sequential estimation algorithms. In these methods, the state-error covariance matrix is replaced by its square root during the propagation and update of the estimate.[1-5] The state error covariance matrix does not appear explicitly and, if it is required, it can be obtained by multiplying the square root covariance by its transpose. Consequently, it will always be semipositive definite.

In the initial formulations,[1-4] the enhanced numerical stability was obtained at the expense of increased computation complexity and an associated increase in computation time. In Ref. 5, a square root measurement update method is proposed which offers potential improvement in the computational efficiency of the square root filtering methods. This efficiency is created by maintaining the square root covariance matrix in triangular form. Following a procedure based on Givens' transformation,[6] an algorithm has been proposed[7] which factors the state-error convariance P into the form $P \equiv UDU^T$, where U is unit upper triangular and D is diagonal. Using the UDU^T factorization eliminates the square root functions present in the algorithms discussed in Refs. 1-5. The measurement incorporation formulation derived for this factorization technique is summarized in the Appendix. The proposed algorithm[7] for propagating the estimate is summarized in the following paragraphs.

The discrete time propagation equation for the state error covariance matrix is

$$\bar{P}_{k+1} = \Phi(t_{k+1},t_k)\hat{P}_k\Phi^T(t_{k+1},t_k) + \Gamma_{k+1} \tag{1}$$

where $\bar{P}_{k+1}$ is the a priori state error covariance matrix at t_{k+1}, $\hat{P}_k$ is the a posteriori state error covariance matrix at t_k, $\Phi(t_{k+1},t_k)$ is the state transition matrix used to map the state from t_k to t_{k+1}, and Γ_{k+1} is the matrix which accounts for the effects of process noise in the interval from t_k to t_{k+1}. The matrices $\Phi(t_{k+1}, t_k)$ and Γ_{k+1} satisfy the following equations:

$$\dot\Phi(t,t_k) = A(t)\Phi(t,t_k) \qquad \Phi(t_k,t_k) = I \tag{2}$$

$$\Gamma_{k+1} = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1},\tau)Q(\tau)\Phi^T(t_{k+1},\tau)\,d\tau \tag{3}$$

where $A(t)$ is a known $n \times n$ time-dependent matrix and $Q(t)$ is the process noise covariance matrix.

The discrete square root time propagation algorithm, based on the UDU^T transformation, can be summarized as follows[7]: form the two matrices

$$W_{K+1} \equiv [\Phi(t_{k+1},t_k)\hat{U}_k \; \vdots \; B_{k+1}] \tag{4}$$

$$\tilde{D}_k = \begin{bmatrix} \hat{D}_k & 0 \\ 0 & Q_k/\Delta t \end{bmatrix} \tag{5}$$

where

$$B_{k+1} \equiv \int_{t_k}^{t_{k+1}} \Phi(t_{k+1},\tau)\mathrm{d}\tau \tag{6}$$

The process noise covariance matrix, Q_k is assumed to be constant in the integration interval $\Delta t \equiv t_{k+1} - t_k$, and Γ_{k+1}, as defined in Eq. (3), is approximated as

$$\Gamma_{k+1} \equiv B_{k+1}(Q_k/\Delta t)B^T_{k+1} \tag{7}$$

Then, the updated factors ($\bar{U}_{k+1}$, $\bar{D}_{k+1}$) are obtained in upper triangular and diagonal forms, respectively, by performing a Modified Weighted Gram-Schmidt orthogonalization on the matrix W_{k+1}, wheres its columns are weighted by the diagonal matrix $\tilde{D}_k$.[7]

The calculation of $\Phi(t_{k+1},t_k)$ requires the integration of $n \times n$ equations in addition to the n-state equations. The determination of B_{k+1} necessitates an $n \times n$ quadrature. Therefore, the total number of equations to be integrated is $2(n \times n)+n$. The UDU^T formulation proposed in Ref. 7 approximates B_{k+1} by an analytical trapezoid-rule integration which eliminates the $n \times n$ quadrature. The error introduced by this approximation can be neglected if the propagation interval $(t_{k+1} - t_k)$ is small. The effect of error accumulated over long prediction intervals, during loss of tracking or data drop-outs, must be considered to ascertain the accuracy of this approximation.

The matrix multiplication, $\Phi\hat{U}_k$, combined with the creation of the augmented matrix W_{k+1}, destroys the triangularity of the square root covariance matrix. The application of the modified Gram-Schmidt orthogonalization procedure is required to retriangularize the UD factors at the time each measurement is processed. The added computational burden of this orthogonalization at each observation point could be eliminated if the square root covariance matrix were propagated without the loss of its triangularity.

In this investigation, a method is proposed which allows the integration of the continuous state-error covariance differential equations in square root form. The derivation follows the approach used in Ref. 8, but the results are based on the $P \equiv UDU^T$ decomposition. The new algorithm can be combined with a triangular measurement update algorithm to obtain a complete square root estimation algorithm for which square roots are avoided. In addition, the effects of state process noise are included without approximation.

Square Root Propagation Equations in Triangular Form

The differential equation for propagating the state-error covariance matrix can be expressed as

$$\dot{\bar{P}}(t) = A(t)\bar{P}(t) + \bar{P}(t)A^T(t) + Q(t) \tag{8}$$

where $\bar{P}(t)$ is the a priori state error covariance matrix, $A(t)$ is the $n \times n$ linearized dynamics matrix, and $Q(t)$ is the process noise covariance matrix. Each of the matrices in Eq. (8) is time-dependent in the general case. However, for simplicity, the time dependence will not be noted specifically in the following discussion.

If the following definitions are used,

$$\bar{P} \equiv \bar{U}\bar{D}\bar{U}^T \qquad \bar{Q} \equiv Q/2 \tag{9}$$

and if the first part of Eq. (9) is differentiated with respect to time and substituted into Eq. (8), the results can be rearranged to form

$$(\dot{\bar{U}}\bar{D} + \bar{U}\dot{\bar{D}}/2 - \bar{Q}\bar{U}^{-T} - A\bar{U}\bar{D})\bar{U}^T + \bar{U}(\bar{D}\dot{\bar{U}}^T + \dot{\bar{D}}\bar{U}^T/2 - \bar{U}^{-1}\bar{Q}^T - \bar{D}\bar{U}^T A^T) = 0 \tag{10}$$

Noting that the first term of Eq. (10) is the transpose of the second term, and making the following definition:

$$C(t) \equiv (\dot{\bar{U}}\bar{D} + \bar{U}\dot{\bar{D}}/2 - \bar{Q}\bar{U}^{-T} - A\bar{U}\bar{D})\bar{U}^T \tag{11}$$

one obtains

$$C(t) + C^T(t) = 0 \tag{12}$$

Relation (12) requires that $C(t)$ be either the null matrix or, more generally, skew symmetric.

Equation (11) can be simplified by selectively carrying out the multiplication of the $-\bar{Q}\bar{U}^{-T}$ term by $\bar{U}^T$ to yield, after terms are rearranged,

$$(\dot{\bar{U}}\bar{D} + \bar{U}\dot{\bar{D}}/2 - A\bar{U}\bar{D})\bar{U}^T = \bar{Q} + C(t) \equiv \tilde{C}(t) \tag{13}$$

Equation (13) defines the differential equations for $\bar{U}$ and $\bar{D}$ to the degree of uncertainty in $C(t)$. Since the unknown matrix $C(t)$ is skew symmetric, there exist $n(n-1)/2$ unknown scalar quantities in Eq. (13). The problem considered here is one of specifying the elements of $C(t)$ so that $\bar{U}$ is maintained in triangular form during the integration of Eq. (13). (The derivation pursued here assumes that $\bar{U}$ is lower triangular and $\bar{D}$ is diagonal, although an algorithm for an upper triangular $\bar{U}$ can be obtained as easily.) The following definitions are made to facilitate the solution to the problem posed above.

$$T \equiv A\bar{U}\bar{D} \qquad M \equiv \dot{\bar{U}}\bar{D} + \bar{U}\dot{\bar{D}}/2 - T \tag{14}$$

With these definitions, Eq. (13) is expressed as

$$M\bar{U}^T = \tilde{C} = \bar{Q} + C(t) \tag{15}$$

Since $\dot{\bar{U}}$ and $\bar{U}$ in Eq. (13) are lower triangular, and since from Eq. (12), $C(t)$ is skew symmetric, several observations can be made regarding Eq. (15). There are $n(n-1)/2$ unknown elements in $\tilde{C}$. The products $\dot{\bar{U}}\bar{D}$ and $\bar{U}\dot{\bar{D}}$ are lower triangular creating $n(n+1)/2$ unknowns. Therefore, the $n \times n$ system of equations (15) has $[n(n-1)/2 + n(n+1)/2] = n \times n$ unknowns which can be determined uniquely.

An expansion of Eq. (15) into matrix elements indicates the method of solution:

$$\begin{bmatrix} M_{11} & -T_{12} & \cdots & -T_{1n} \\ M_{21} & M_{22} & \cdots & -T_{2n} \\ \vdots & & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & M_{nn} \end{bmatrix} \begin{bmatrix} 1 & \bar{U}_{21} & \cdots & \bar{U}_{n1} \\ \cdot & 1 & \cdots & \bar{U}_{n2} \\ \vdots & & \ddots & \vdots \\ 0 & \cdot & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \bar{q}_{11} & -C_{21} & \cdots & -C_{n1} \\ C_{21} & \bar{q}_{22} & \cdots & -C_{n2} \\ \vdots & & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & \bar{q}_{nn} \end{bmatrix} \tag{16}$$

In Eq. (16), $\bar{Q}$ is assumed to be a diagonal matrix with elements $\bar{q}_{ii}=q_{ii}/2$ $(i=1,...,n)$. (This assumption can be generalized to allow other nonzero terms in the $\bar{Q}$ matrix with only a slight increase in algebraic complexity.) Each row of the upper triangular portion of the $\bar{C}$ matrix in Eq. (16) is determined as the product of the corresponding row of the M matrix with the appropriate column of the $\bar{U}^T$ matrix. After an upper triangular row of C is determined, the condition from Eq. (12) that $C_{ij}=-C_{ji}$ $(i=1,...,n;\ j=1...,i-1)$ is invoked to evaluate the corresponding lower triangular column of C. Then, a column of the lower triangular elements of M can be evaluated. Once the elements of the M matrix are determined, the new row of the upper triangular C elements can be computed along with a column of the $\dot{\bar{U}}$ and $\dot{\bar{D}}$ elements. This process is repeated until all $\dot{\bar{U}}$ and $\dot{\bar{D}}$ values are determined. The implementation of this approach proceeds as follows. From Eqs. (13) and (14) one can write

$$M+T=\dot{\bar{U}}\bar{D}+\bar{U}\dot{\bar{D}}/2 \tag{17}$$

The expansion of Eq. (17) in summation notation gives

$$M_{ij}+T_{ij}=\sum_{k=1}^{n}\dot{\bar{U}}_{ik}\bar{d}_{kj}+\sum_{k=1}^{n}\frac{\bar{U}_{ik}\dot{\bar{d}}_{kj}}{2} \quad (i=1,...,n;\ j=1,...,i) \tag{18}$$

But, since $\bar{D}$ is diagonal, Eq. (18) becomes

$$M_{ij}+T_{ij}=\dot{\bar{U}}_{ij}\bar{d}_{jj}+\bar{U}_{ij}\dot{\bar{d}}_{jj}/2 \qquad (i=1,...,n;\ j=1,...,i) \tag{19}$$

for $i=j$, $\bar{U}_{ij}\equiv 1$, and $\dot{\bar{U}}_{ij}\equiv 0$. Therefore, Eq. (19) becomes

$$\dot{\bar{d}}_{ii}=2(M_{ii}+T_{ii}) \qquad (i=1,...,n) \tag{20}$$

For $i>j$, Eq. (19) is rearranged to obtain the differential equation

$$\dot{\bar{U}}_{ij}=(M_{ij}+T_{ij}-\bar{U}_{ij}\dot{\bar{d}}_{jj}/2)/\bar{d}_{jj} \quad (i=2,...,n;\ j=1,...,i-1) \tag{21}$$

Equations (20) and (21) are the forms of the differential equations to be employed in the derivative routine of a numerical integrator. The elements T_{ij} and M_{ij} are computed as defined in Eq. (14). The pertinent equations can be combined to obtain the following algorithm.

Triangular Square Root Propagation Algorithm

Given the elements of the square root state-error covariance in lower triangular $\bar{U}\bar{D}\bar{U}^T$ form, $\bar{Q}\equiv Q/2$ and $A(t)$, the differential equations $\dot{\bar{U}}_{ij}$ and $\dot{\bar{d}}_{ii}$ can be computed as follows:

$$T_{ij}=\sum_{k=j}^{n}A_{ik}\bar{U}_{kj}\bar{d}_{jj} \qquad (i=1,...,n;\ j=1,...,n) \tag{22}$$

$$\bar{C}_{ij}=\sum_{k=1}^{i}M_{ik}\bar{U}_{jk}-\sum_{k=i+1}^{j}T_{ik}\bar{U}_{jk} \qquad (i=1,...,n;\ j=i+1,...,n) \tag{23}$$

$$M_{ii}=\bar{q}_{ii}-\sum_{k=1}^{i-1}M_{ik}\bar{U}_{ik} \qquad (i=1,...,n) \tag{24}$$

$$M_{ij}=-\bar{C}_{ji}-\sum_{k=1}^{j-1}M_{ik}\bar{U}_{jk} \quad (i=2,...,n;\ j=1,...,i-1) \tag{25}$$

$$\dot{\bar{d}}_{ii}=2(M_{ii}+T_{ii}) \qquad (i=1,...,n) \tag{26}$$

$$\dot{\bar{U}}_{ij}=(M_{ij}+T_{ij}-\bar{U}_{ij}\dot{\bar{d}}_{jj}/2)/\bar{d}_{jj} \quad (i=2,...,n;\ j=1,...,i-1) \tag{27}$$

The propagation algorithm summarized in Eqs. (22-27) can be combined with the algorithm for incorporating an observation[7] to obtain a complete sequential estimation algorithm in which the covariance matrices $\hat{P}$ and $\bar{P}$ are replaced by the factors $(\hat{U},\hat{D})$ and $(\bar{U},\bar{D})$, respectively. The algorithm given in the Appendix assumes that only a single scalar observation is processed at each observation epoch; however, the algorithm is applicable to the case of multiple observations at a given epoch if the observation errors are assumed to be uncorrelated.

Numerical Comparison

In the following numerical example, the two methods for propagation of the UDU^T factored covariance matrix are compared to determine the relative computation speed and estimation accuracy. As a basis for determining their absolute performance, numerical results are obtained with the conventional Extended Kalman-Bucy Filter using both Eqs. (1) and (8) as the bases for propagating the state error covariance matrix. The numerical comparisons are made by using each of the algorithms to process a set of simulated Global Positioning System (GPS) range and range-rate observations obtained by the LANDSAT-D spacecraft.[9] A detailed discussion of the GPS and the associated navigation measurements is given in Ref. 10. Since the range measurement will require a precise measurement of the time interval between signal transmission from one of the GPS satellites to reception at the LANDSAT-D spacecraft, the clock error must be modeled and included as part of the overall state vector. The primary clock errors are the bias and drift.

The dynamic model used for the motion of LANDSAT-D and the associated model of the satellite's clock are combined to obtain a filter model which contains nine state variables. The nine components of the state vector are: position ($\bar{r}$; 3×1), velocity ($\bar{v}$; 3×1), clock bias (b), clock drift (n), and clock drift model correlation parameter (β). The differential equations defining these parameters in the filter are

$$\dot{\bar{r}}=\bar{v} \tag{28}$$

$$\dot{\bar{v}}=\bar{g}+\bar{a}_d+\bar{\xi}_v \tag{29}$$

$$\dot{b}=n \tag{30}$$

$$\dot{n}=\beta n+\xi_n \tag{31}$$

$$\dot{\beta}=\xi_\beta \tag{32}$$

The stochastic processes $\bar{\xi}_v$, ξ_n and ξ_β are white noise forcing functions which are assumed to have the following statistics:

$$E[\xi(t)]_i=0 \qquad E[\xi(t)\xi^T(\tau)]_i=Q_i(t)\delta(t-\tau) \tag{33}$$

where the subscript i indicates the appropriate member of the set $\{\bar{v},n,\beta\}$ and $\delta(t-\tau)$ is the Dirac delta function.

For computational efficiency, the filter assumes simple models for the gravitational acceleration $\bar{g}$ and for the atmospheric drag acceleration $\bar{a}_d$. The geopotential model adopted for the filter is obtained by truncating the Goddard Earth Model (GEM7)[11] to the fourth degree and order. The drag acceleration is calculated as

$$\bar{a}_d=-d\gamma v_{\rm rel}\bar{v}_{\rm rel} \tag{34}$$

where the atmospheric density γ is approximated by the exponential model

$$\gamma=\gamma_0 e^{-k(h-h_0)} \tag{35}$$

The values of the drag model parameters assumed for this investigation are: $h_0=840{,}000$ m, $\gamma_0=5.74\times 10^{-14}$ kg/m^3, $k=7.58\times 10^{-6}$m^{-1}, and $d=1.18\times 10^{-2}$ m^2/kg.

Table 1 Initial conditions for numerical simulations

State	State covariance	Noise covariance
$X(1) = 7.046 \times 10^6$ m	$P(1,1) = 6 \times 10^6$ m^2	$Q(1,1) = 0$
$X(2) = -5.433 \times 10^6$ m	$P(2,2) = 6 \times 10^6$ m^2	$Q(2,2) = 0$
$X(3) = -6.120 \times 10^5$ m	$P(3,3) = 6 \times 10^6$ m^2	$Q(3,3) = 0$
$X(4) = 5.562 \times 10^2$ m/s	$P(4,4) = 1 \times 10^4$ (m/s)^2	$Q(4,4) = 1 \times 10^{-6}$ m^2/s^3
$X(5) = -1.116 \times 10^3$ m/s	$P(5,5) = 1 \times 10^4$ (m/s)^2	$Q(5,5) = 1 \times 10^{-6}$ m^2/s^3
$X(6) = 7.388 \times 10^3$ m/s	$P(6,6) = 1 \times 10^4$ (m/s)^2	$Q(6,6) = 1 \times 10^{-6}$ m^2/s^3
$X(7) = 2.998 \times 10^1$ m	$P(7,7) = 3.600 \times 10^3$ m^2	$Q(7,7) = 0$
$X(8) = 5.996 \times 10^{-1}$ m/s	$P(8,8) = 6. \times 10^0$ (m/s)^2	$Q(8,8) = 1 \times 10^{-4}$ m^2/s^3
$X(9) = 5.550 \times 10^{-4}$ s^{-1}	$P(9,9) = 1 \times 10^{-5}$ s^{-2}	$Q(9,9) = 1 \times 10^{-7}$ s^{-3}

Table 2 Numerical algorithm comparison for the modified Euler integrator (step size = 6 s)

Algorithm	Meas. update, ms	Time update, ms	Norm. time	Total update, ms	Accuracy	
					Rms pos., m	Rms vel., m/s
$EKF(\dot{P})$	7.246	38.608	1.000	45.855	113.1	0.431
$EKF(\dot{\Phi})$	8.057	42.792	1.108	50.849	120.3	0.432
$UDU^T(\dot{U}\dot{D})$	8.045	51.131	1.324	59.176	111.4	0.433
$UDU^T(\dot{\Phi})$	8.216	61.296	1.588	69.512	117.6	0.432

Table 3 Numerical algorithm comparison for the modified Euler integrator (step size = 3 s)

Algorithm	Meas. update, ms	Time update, ms	Norm. time	Total update, ms	Accuracy	
					Rms pos., m	Rms vel., m/s
$EKF(\dot{\Phi})$	7.981	64.688	1.000	72.669	148.5	0.442
$EKF(\dot{P})$	8.341	75.224	1.163	83.565	146.8	0.441
$UDU^T(\dot{\Phi})$	7.867	82.872	1.281	90.739	146.8	0.441
$UDU^T(\dot{U}\dot{D})$	6.972	102.831	1.590	109.803	146.0	0.440

The estimates of the clock bias b and drift n are used to predict the true time of the user's clock t by the equation

$$t = T - b_0 - n_0(t - t_0) \tag{36}$$

where the subscript zero indicates the epoch of the last estimate of the parameters. The quantity T is the time as indicated by the LANDSAT-D clock.

The observations used for this study are pseudo range and pseudo range-rate measurements as observed by the LAND SAT-D satellite using the six satellites of the Phase I GPS constellation.[10] The computer software used to simulate the observations as well as further details on the simulation procedure are discussed in Ref. 12.

The models used to generate the simulated measurements have the form

$$Y_\rho \equiv \rho + (b_l - b_s)c + \xi_\rho \tag{37}$$

$$Y_{\dot{\rho}} \equiv \dot{\rho} + (n_l - n_s)c + \xi_{\dot{\rho}} \tag{38}$$

where ρ and $\dot{\rho}$ are the true values of range and range-rate between LANDSAT-D and the given GPS satellite; Y_ρ and $Y_{\dot{\rho}}$ are the measured values of the range and range-rate; b_l, n_l, b_s, and n_s are the biases and drifts in the satellite clock and in the GPS clocks, respectively; and c is the speed of light.[10]

The measurement Y_ρ and $Y_{\dot{\rho}}$ is processed by the LAND-SAT-D navigation filter using the model

$$Y_\rho = \rho + cb + \xi_\rho \tag{39}$$

$$Y_{\dot{\rho}} = \dot{\rho} + cn + \xi_{\dot{\rho}} \tag{40}$$

That is, in the filter model the GPS clocks are assumed to be perfect, and the total time error is assumed to be contained in the LANDSAT-D satellite clock. The interval between observations is assumed to be 6 s, and observations from only one GPS satellite can be obtained during any 6 s interval. The observations from the visible satellites are processed sequentially.

The GPS satellites are assumed to be in circular orbits ($e = 0$) about a point mass earth with inclinations of 63 deg and periods of 12 h (43,200 s). Three satellites are equally spaced on each of two orbital planes. The orbital elements are referenced to a coordinate system whose xy-plane is the earth's equator and whose xz-plane lies along the Greenwich meridian.

The epoch condition for LANDSAT-D was chosen so that the resulting simulated observations would accurately reflect the possible extremes of GPS satellite visibility. The epoch elements chosen are $a \equiv 7.086901 \times 10^6$m, $e \equiv 0.0001$, $I \equiv 98.181$ deg, $\Omega \equiv 354.878$ deg, $\omega \equiv 180$ deg, and f (true anomaly) $\equiv -185$ deg. The elements are specified at a GPS system time $t = 0$. The epoch elements for GPS are specified at a system time of -7200 s. The difference in initial epochs is included in the filter program's update of user and GPS states.

The values of the LANDSAT-D and GPS clock phase and frequency errors are simulated as the sum of three different error sources: a noise-free phase error with a polynomial form, an error due to exponentially correlated frequency noise, and a random walk bias error. The exact form of the error models and the coefficients used in the models are given in Ref. 12.

The numerical simulations of the filter performance were made with the initial conditions shown in Table 1, with the off-diagonal terms of the state-error covariance and noise covariance matrices set to zero initially.

The data in Tables 2 and 3 show the relative performance of the four algorithms when each processed a simulated observation data set with a duration of approximately 10,000 s. The five columns in each of the tables are, respectively: 1) the total CPU (central processing unit) time required to perform measurement updates of the state and covariance, 2) the total CPU time required for propagating the state and covariance, 3) the total CPU propagation time normalized by the fastest CPU propagation time, 4) the total CPU time required for propagation and measurement updates (the sum of columns 1 and 2), and 5) the rms of the position magnitude errors and velocity magnitude errors over the duration of the simulation. All CPU times are listed in ms. Position errors are given in m and velocity errors are given in m/s. The algorithms are ranked in the tables in order of increasing total computation time.

Time propagation is performed with a fixed-step modified Euler integrator, which requires two function evaluations per step. The integration step size for the data of Table 2 is 6 s, equal to the time interval between GPS observations. The data of Table 3 result from integration with a 3 s step size.

The relations for the propagation of the time bias and drift, as approximated by Eqs. (30-32) have simple analytic solutions. This allows certain elements of the state transition matrix to be updated analytically. The implementation of the $UDU^T(\dot{\Phi})$ and $EKF(\dot{\Phi})$ algorithms has taken advantage of this simplification to reduce the number of numerically integrated differential equations below the theoretical value of n^2. The high degree of coupling in the covariance differential equations for the $(\dot{P})$ and $(\dot{U},\dot{D})$ algorithms does not permit a convenient reduction in the integration vector size. A total of $n(n+1)/2$ covariance equations has been integrated in the simulations described here.

While the programming effort required to implement the $(\dot{U},\dot{D})$ formulation will be greater because of the recursive nature of Eqs. (22-27), fewer computer storage locations will be required to execute this algorithm since the total number of equations involved in Eqs. (22-27) is $n(n+1)/2$. This value compares with the $(n\times n)$ computer memory locations required to integrate and store the $\dot{\Phi}$ equations. Since there are some zeros in the $\dot{\Phi}$ equation, one can reduce the storage requirements of this method at the expense of added programming complexity. Further comparisons of the computer storage location requirements for these two algorithms are given in Ref. 12.

The numerical results shown in Table 2 indicate that the $(\dot{U},\dot{D})$ algorithm is competitive with both the $UDU^T(\dot{\Phi})$ and conventional formulations in terms of CPU times and estimation accuracy for this filtering problem. The $(\dot{U},\dot{D})$ method is faster than the $UDU^T(\dot{\Phi})$ algorithm for the 6 s integration interval. Its position estimation error is lower than that for any of the other algorithms. For the 3 s step size results in Table 3, the $(\dot{U},\dot{D})$ algorithm remains competitive in terms of estimation accuracy, but is no longer as fast as the $UDU^T(\dot{\Phi})$ algorithm. With the decrease in integration step size, the number of expensive $(\dot{U},\dot{D})$ function evaluations has increased, but the number of time-consuming orthogonalizations in the $UDU^T(\dot{\Phi})$ algorithm remains the same. This factor causes the $UDU^T(\dot{\Phi})$ algorithm to have a faster computation time in this case. Again, for the 3 s results, the $(\dot{U},\dot{D})$ formulation yields the most accurate position estimate.

Both UDU^T methods require higher total computation times than the *EKF* algorithms. This is not an unexpected result as square root filter formulations often incur computation time penalities as the price for increased numerical stability.

These numerical results were generated on a CDC6600 computer system. The relative performance of the algorithms will vary as a function of the computer system being used, the dynamic model assumed by the filter, the method and order of numerical integration, and the integration step size. The influence on performance of the latter two factors can be seen in the numerical results given in Ref. 12. The $(\dot{U},\dot{D})$ method becomes more efficient as the total number of function evaluations within a given integration interval is decreased. Therefore, the choice of the time propagation method should depend on the formulation of the specific problem under consideration and on the supporting algorithms and computer system used to perform the calculations.

Conclusions

Based on the results presented in the previous discussion, it is concluded that, for the example problem considered here, the $(\dot{U},\dot{D})$ algorithm is more efficient than the (UDU^T) algorithm based on the $\dot{\Phi}$ propagation. Furthermore, the estimate obtained with the $(\dot{U},\dot{D})$ formulation was more accurate than the estimate obtained by either the conventional *EKF* estimation algorithms or the $UDU^T(\dot{\Phi})$ algorithm. The performance of the algorithms will be dependent on the computer architecture and software, the dynamic model assumed for the filter and the method used to perform the numerical integrations, and will vary as these factors change.

Appendix

The measurement update algorithm for the UDU^T factorization[7] has the following form. Using the observation $Y_{k+1}=G(X_{k+1},t_{k+1})$, calculate:

$$H_{k+1}=[\partial G(\bar{X}_{k+1},t_{k+1})/\partial \bar{X}_{k+1}] \qquad \text{(A1)}$$

For $i=1\rightarrow n$,

$$\tilde{F}_i=H_i+\sum_{k=i+1}^{n} H_k\bar{U}_{ki} \qquad \text{(A2)}$$

$$V_i=\bar{d}_i\tilde{F}_i \qquad \text{(A3)}$$

Set $\beta_{n+1}=R_{k+1}$ (where R_{k+1} is the measurement noise) and calculate:

$$\beta_i=\beta_{i+1}+V_i\tilde{F}_i \qquad (i=n\rightarrow 1) \qquad \text{(A4)}$$

Calculate diagonal covariance elements:

$$\hat{d}_i=\bar{d}_i+\beta_{i+1}/\beta_i \qquad (i=n\rightarrow 1) \qquad \text{(A5)}$$

$$\alpha=\beta_1 \qquad \text{(A6)}$$

For $i=2\rightarrow n$ and $j=1\rightarrow i-1$, calculate:

$$P_j=\tilde{F}_j/\beta_{j+1} \qquad \text{(A7)}$$

$$B_{ij}=V_i+\sum_{k=j+1}^{i-1}\bar{U}_{ik}V_k \qquad \text{(A8)}$$

$$\hat{U}_{ij}=\bar{U}_{ij}-B_{ij}P_j \qquad (i=2\rightarrow n;\ \ j=1\rightarrow i-1) \qquad \text{(A9)}$$

Compute residual:

$$y_{k+1}=Y_{k+1}-G(\bar{X}_{k+1},t_{k+1}) \qquad \text{(A10)}$$

Calculate gain and update state:

$$\tilde{K}_i=B_{i1}+\bar{U}_{i1}V_1 \qquad (i=1\rightarrow n) \qquad \text{(A11)}$$

$$\hat{X}_i=\bar{X}_i+\tilde{K}_i y_{k+1}/\alpha \qquad (i=1\rightarrow n) \qquad \text{(A12)}$$

Acknowledgments

This research was supported by NASA Goddard Space Flight Center under Grant No. NSG 5154.

References

[1] Battin, R.H., *Astronautical Guidance*, McGraw-Hill, New York, 1964, pp. 338-389.

[2] Bellantoni, J.F. and Dodge, K.W., "A Square Root Formulation of the Kalman-Schmidt Filter," *AIAA Journal*, Vol. 5, July 1967, pp. 1309-1314.

[3] Kaminsky, P.G., Bryson, A.E., and Schmidt, S., "Discrete Square Root Filtering: A Survey of Current Techniques," *IEEE Transactions on Automatic Control*, Vol. AC-16, Dec. 1971, pp. 727-736.

[4] Andrews, A., "A Square Root Formulation of the Kalman Covariance Equations," *AIAA Journal*, Vol. 6, June 1968, pp. 1165-1166.

[5] Carlson, N.A., "Fast Triangular Formulation of the Square Root Filter," *AIAA Journal*, Vol. 11, Sept. 1973, pp. 1239-1265.

[6] Gentleman, W.M., "Least Squares Computations by Givens Transformations without Square Roots," *Journal of the Institute of Mathematical Applications*, Vol. 12, 1973, pp. 329-336.

[7] Bierman, G.J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, 1976, pp. 124-133.

[8] Tapley, B.D. and Choe, C.Y., "An Algorithm for Propagating the Square Root Covariance Matrix in Triangular Form," *IEEE Transactions on Automatic Control*, Vol. AC-21, Feb. 1976, pp. 122-123.

[9] Wooden, W.H. and Dunham, J.B., "Simulation of Autonomous Satellite Navigation with the Global Positioning System," AIAA Paper 78-1429, Palo Alto, Calif., Aug. 1978.

[10] Kruczynski, L., "Global Positioning System Navigation Algorithms," Applied Mechanics Research Laboratory, Dept. of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, Tex., Report No. AMRL 1078, May 1976.

[11] Wagner, C.A., et al., "Improvement in the Geopotential Derived from Satellite and Surface Data (GEM 7 and 8)," NASA-Goddard Space Flight Center, Greenbelt, Md., Report. No. GSFC X-921-76-20, Jan. 1976.

[12] Tapley, B.D., Peters, J.G., and Schutz, B.E., "A Comparison of Square Root Estimation Algorithms for Autonomous Satellite Navigation," Institute for Advanced Study in Orbital Mechanics, Dept. of Aerospace Engineering and Engineering Mechanics, University of Texas Austin, Tex., Report No. TR79-1, March 1980.