Accuracy

Misleading claims are made about the accuracy of the $\dot{U}$-$\dot{D}$ approach. The *U-D* differential equations form a complex nonlinear system, and nothing is stated in Ref. 1 about the stability of these equations, except that their use resulted in excellent (but, in this author's opinion, somewhat questionable) results in a single brief simulation application. It is important that one investigate whether these equations are stiff, and determine from a numerical stability and algorithm accuracy point of view how integration accuracy is influenced by the dynamics, measurements (which, while not explicitly a part of the numerical integration, do influence the *U-D* factors), and the process noise intensity elements, Q_c. Tapley and Peters tested their *U-D* differential equation algorithms on a CDC 6600 computer. Because of this machine's long wordlength, their results do not necessarily reflect the relative accuracy between the Gram-Schmidt discrete-time update and the Tapley-Peters continuous-time approach. What their results seem to show is the effect of numerical integration error on the discrete-time transition matrix and *U-D* differential equation approaches. That the integration results of Ref. 1 are questionable can be seen by comparing the rms position errors in Tables 2 and 3. With a 6 s integration step, the $\dot{U}$-$\dot{D}$ rms position error listed is 111.4 m. When the integration step is decreased to 3 s, the results worsen (i.e., the estimate errors increase) to 146.0 m, a 30% accuracy degradation.

A further point concerning accuracy is that the massive amount of literature on transition matrix computation, summarized in Ref. 7, has culminated in accurate, reliable, and efficient methods for such computation. The $\dot{U}$-$\dot{D}$ equations, being new, nonlinear, and having a structure not previously studied, have not yet been analyzed. Consequently, the sensitivity of the solutions to stepsize, system stability, measurement update rate, integration algorithm, and *U-D* singularities all need to be studied.

Concluding Remarks

The $\dot{U}$-$\dot{D}$ continuous-time propagation algorithm described in Ref. 1 appears to be a promising approach to the continuous-time filter problem. The authors have, however, exaggerated the merits of their approach. A more meaningful demonstration of their algorithm would be for them to apply their LANDSAT-D test problem on a 16- or 32-bit wordsize computer. There is a need for a such demonstration because, besides not demonstrating the numerical consistency of their approach, Tapley and Peters assert that their continuous-time propagation algorithm is more accurate than is the discrete-time *U-D* Gram-Schmidt algorithm. To add perspective, we point out that discrete-time *U-D* algorithms have been in use for the past several years, including a number of ill-conditioned applications and that no cases of numerical failure or significant accuracy degradation have yet been reported. On the other hand, such positive results have not been the case for the continuous-time Tapley-Choe covariance square root algorithm, reported in Ref. 5, which is quite analogous to the *U-D* approach that is discussed herein. The point to be made is that accuracy analysis studies of the continuous-time Tapley-Choe covariance square root and the discrete-time Carlson triangular square-root† algorithm were carried out for the phase 1 development of the Global Positioning System. It was concluded that the square root covariance differential equation was unstable for this application and, based upon its performance, it was decided that discrete-time factorization should be used.

Our goal in this correspondence is to highlight strengths of the discrete model and the associated *U-D* time propagation that were overlooked in Ref. 1, and to correct some overzealous claims made about the merits of the continuous-time *U-D* propagation process. Despite the flavor of our comments, we are enthusiastic about the Tapley-Peters continuous-time propagation algorithm. It may well be that in certain applications and/or studies the continuous-time approach will be more effective, and/or give greater engineering insight than do the discrete-time approaches. Much remains to be done, including study of the *U-D* differential equations to see how they simplify to accommodate Markov colored noise and bias parameters, development of methods for directly computing *U-D* steady-state matrix factors, which numerical integration methods are most efficiently applied to the *U-D* differential equations, etc.

†The Carlson algorithm,[8] is a less efficient but quite similar predecessor of the *U-D* approach.

References

[1] Tapley, B.D. and Peters, J.G., "Sequential Estimation Algorithm Using a Continuous UDU^T Covariance Factorization," *Journal of Guidance and Control,* Vol. 3, July-August 1980, pp. 326-331.

[2] Thornton, C.L. and Bierman, G.J., "Gram-Schmidt Algorithms for Covariance Propagation," *International Journal of Control,* Vol. 25, No. 2, 1977, pp. 243-260.

[3] Bierman, G.J., *Factorization Methods for Discrete Sequential Estimation,* Academic Press, New York, 1977, Chaps. VI, VII.

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

[5] 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.

[6] Thornton, C.L. and Bierman, G.J., "UDU^T Covariance Factorization For Kalman Filtering," in *Advances In Control and Dynamic Systems,* Vol. 16, edited by C.T. Leondes, Academic Press, New York, 1980, pp. 177-248.

[7] Moler, C.B. and Van Loan, C., "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review,* Vol. 20, Oct. 1978, pp. 801-836.

[8] Carlson, N.A., "Fast Triangular Formulation of the Kalman Covariance Equations," *AIAA Journal,* Vol. 15, Sept. 1977, pp. 1259-1265.

AIAA 81-4173

Reply by Authors to G.J. Bierman

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

THE authors agree with many of Bierman's comments regarding Ref. 1. In fact, several of the points mentioned by Bierman are discussed in more detail in Refs. 2 and 3. However, there are several points made by Bierman with which the authors do not agree. These points are summarized as follows.

Equivalence

In his initial remarks, Bierman illustrates the equivalence of the continuous-form linear system model

$$\dot{x} = Ax + B_c \xi \qquad x(t_0) \in N(\bar{x}_0, P_0) \tag{1}$$

with the discrete-form system model

$$x(t_{j+1}) = \Phi(t_{j+1}, t_j) x(t_j) + B_D w_j \tag{2}$$

Received Dec. 29, 1980.

*Professor, Department of Aerospace Engineering and Engineering Mechanics. Associate Fellow AIAA.

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

In the above equations, A and B_c can be time dependent, $\xi(t)$ is white Gaussian noise with diagonal intensity $Q_c = Q_c(t)$, and $\Phi(t_{j+1}, t_j)$ is the solution of the differential equation

$$\dot{\Phi}(t,t_j) = A(t)\Phi(t,t_j) \qquad \Phi(t_j,t_j) = I$$

$\{B_D w_j\}$ is a white Gaussian sequence defined as

$$B_D w_j \triangleq \int_{t_j}^{t_{j+1}} \Phi(t_{j+1},s) B_c(s) \xi(s) \mathrm{d}s \tag{3}$$

The only assumption implied by the solution of Eq. (2) is the formal definition given by Eq. (3).

The equivalence of the continuous covariance update,

$$\dot{P}(t) = A(t)P(t) + P(t)A^T(t) + Q_c(t) \qquad P(t_j) = \hat{P}_j(t_j|t_j) \tag{4}$$

with the discrete covariance update,

$$\hat{P}(t_{j+1}|t_j) = \Phi(t_{j+1},t_j)\hat{P}(t_j|t_j)\Phi^T(t_{j+1},t_j) + B_D Q_D B_D^T \tag{5}$$

can be obtained by the definition

$$B_D Q_D B_D^T \triangleq \int_{t_j}^{t_{j+1}} \Phi(t_{j+1},s) B_c(s) Q_c(s) B_c^T(s) \Phi^T(t_{j+1},s) \mathrm{d}s \tag{6}$$

Given the definitions in Eqs. (3) and (6), the authors admit the formal equivalence of the discrete and continuous forms of both state update and covariance update equations (1), (2), (4) and (5). However, in many applications, the formal equivalence implied by Eqs (3) and (6) cannot be achieved without carrying out the integration therein at each update, and by then performing a square root decomposition. Such a procedure would negate the advantages of the approach suggested by Bierman.

As explained in Ref. 4, there are two approaches to calculating the integral in Eq. (6). For the elements of the general transition matrix, $\Phi(t_{j+1}, t_s)$, which have analytic time-dependent solutions, Eq. (6) can be solved exactly. Although they may not be convenient to use computationally, proper *U-D* factored elements can be derived and included in a modified weighted Gram-Schmidt orthogonalization. The derivation of these factors for a first-order Gauss-Markov model is illustrated in Ref. 4.

However, when there are no simple analytical expressions for the elements of the transition matrix, some form of numerical approximation must be made to obtain an efficient realization of Eq. (6). Approximations are made as follows (the derivation follows Ref. 4).

The white noise process, $\xi(s)$, is approximated by a piecewise constant white sequence, ξ_j. Therefore, Eq. (3) becomes

$$B_D w_j \cong \left(\int_{t_j}^{t_{j+1}} \Phi(t_{j+1},s) B_c(s) \mathrm{d}s\right) \xi_j \tag{7}$$

Now, the computed covariance of $B_D w_j$ becomes, approximately,

$$B_D E\{w_j w_j^T\} B_D^T = B_D Q_D B_D^T$$

$$\cong \int_{t_j}^{t_{j+1}} \Phi(t_{j+1},s) B_c(s) \mathrm{d}s Q_j \int_{t_j}^{t_{j+1}} B_c^T(\tau) \Phi^T(t_{j+1},\tau) \mathrm{d}\tau \tag{8}$$

where $E\{\xi_j \xi_k^T\} = Q_j \delta_{jk}$. In Eq. (8), it can be shown (see Ref. 5) that $Q_j = Q_c/\Delta t$, where $\Delta t = t_{j+1} - t_j$.

Defining the first integral to be A_j, Eq. (8) can be written as

$$B_D Q_D B_D^T \cong A_j Q_j A_j^T \tag{9}$$

Given the *U-D* factors, U_j and D_j, of the a posteriori covariance, $\hat{P}(t_j|t_j)$, the factors of the a priori covariance, $\hat{P}(t_{j+1}|t_j) \equiv U_{j+1} D_{j+1} U_{j+1}^T$, are obtained[4] applying the modified weighted Gram-Schmidt algorithm to the array:

$$[\Phi(t_{j+1},t_j) U_j \,|\, A_j] \tag{10}$$

where the columns are weighted by diag (D_j, Q_j), and $Q_j = Q_c/\Delta t$. Note, however, that achieving the form of Eq. (9), requires that $E\{\xi(t)\xi^T(\tau)\} \cong Q_j \delta(t-\tau)$, where Q_j is constant over Δt, and that the integral in Eq. (6) can be separated into the two integrals given in Eq. (8). Even if the piecewise constant approximation is satisfied for the process noise, the factorization of the integral in Eq. (6) to obtain the form given by Eq. (8) cannot be accomplished analytically for many applications.

A second approximation is required to implement Eq. (10). This occurs in the evaluation of

$$A_j \equiv \int_{t_j}^{t_{j+1}} \Phi(t_{j+1},s) B_c(s) \mathrm{d}s$$

Bierman has proposed a solution to this integral based on the trapezoid rule, e.g.,

$$A_j \cong [B_c(t_{j+1}) + \Phi(t_{j+1},t_j) B_c(t_j)] \quad \Delta t/2 \tag{11}$$

Defining the first factor on the right-hand side of Eq. (11) as V_j, Eq. (10) becomes

$$[\Phi(t_{j+1},t_j) U_j \,|\, V_j \Delta t/2] \tag{12}$$

Although this second approximation is not essential, an efficient application of the *UDU* algorithm, as suggested in Eq. (12), necessitates some such approximation. The continuous update algorithm,[1] does not require 1) the assumption implied by Eq. (11), 2) the assumption of a piecewise constant process noise, or 3) that the integral in Eq. (6) be factorable.

Our intent in this discussion is not to detract from the validity and usefulness of the approximations made in the discrete update form. In many situations, the approximations offer more than the required computational accuracy. It is our intent, however, to show that the discrete algorithm, when employed with dynamic systems involving nonanalytic transition matrix equations, does require approximations for efficient computation. Contrary to Bierman's remarks, the continuous update form, based on the direct integration of P or *U-D* equations, is not identical to the discrete *U-D* algorithm as proposed by Bierman.

Efficiency

Bierman states that the numerical results in Table 2 of Ref. 1 indicate that the *U-D* measurement update is 13% slower than the conventional Kalman formulation. This appears to be a reference to the relative efficiency of the authors' codes for the *U-D* and Kalman measurement update segments. Indeed, if one compares the lowest Kalman measurement update time and the highest *U-D* measurement update time in Table 2, the 13% value is valid. However, if one considers all recorded measurement update times in Tables 1 and 2 (the measurement update being independent of integration step size), the *U-D* times are not consistently larger. In fact, in Table 2, they are smaller. The measurement update computation times are relatively small when compared to the computation times for propagation and are subject to some statistical variation due to the way they are calculated on the CDC computer. Coding implementation certainly is a significant factor in determining relative algorithm efficiency. However, there is little evidence that the *U-D* update code used in these tests is significantly more cumbersome than the Kalman formulation.

In general, the computation efficiency of square root or factored algorithms relative to the Kalman formulation is problem dependent. In some instances, square root algorithms have indeed incurred large penalties in computation efficiency as a price for their increased stability. The Potter formulation used in the Apollo navigation computer is such an example. However, even an efficiency penalty of 10-15%, regarded as "inconsequential" by Bierman, may be significant in many "real-time" applications Acceptable relative efficiency in a specific situation is purely a matter of judgement.

Ultimate achievable efficiency need not be the most important reason for selecting an algorithm, even if a particular algorithm can be made arbitrarily efficient by the exploitation of a triangular structure, sparseness, etc. As an example, a standard Kalman filter implementation has been chosen for the Space Shuttle navigation system specifically to take advantage of a computer language optimized to perform matrix manipulations. Any level of efficiency achievable by a square root or factored algorithm will not be satisfactory if a matrix formulation of the algorithm is required.

With regard to specific points made by Bierman:

1) So far, the structure of the *U-D* differential equations has not encouraged the use of partial analytic propagation. However, by reordering the state equation, there is no theoretical reason why partial analytic solutions cannot be used in the continuous *U-D* formulation in problems of the nature of those discussed in Ref. 1. The results reported in Ref. 1 do, however, make use of all possible analytic updates in the transition matrix computation. In addition, it should be noted that, for recursive computations, the use of analytic solutions does not necessarily lead to the most efficient implementation.

2) Time invariance of the dynamics does allow the transition matrix to be calculated once, only, in the discrete *U-D* form. Such an approach is not possible with the directly integrated *U-D* equations. However, the condition of time invariance is not applicable to either the Global Positioning System (GPS) problem, considered in Ref. 1, or to most other aerospace applications since the dynamics are usually time dependent.

3) Subsystem decoupling is possible with the transition matrix update; it is not possible with the *U-D* equations. The numerical results presented in Ref. 1 did take advantage of element decoupling in integrating the transition matrix equations.

4) When the same dynamic model is used several times, as in covariance analysis, the transition matrix update form is advantageous. Such uses were not addressed in Ref. 1, and the reusability property of variational equations does not apply in navigation simulations, such as reported in Ref. 1, where the extended form of the Kalman algorithm is used.

Accuracy

The topics of stability and equation stiffness were not addressed in the preliminary investigation reported in Ref. 1. Nor was a detailed study undertaken to ascertain the change in integration accuracy as a function of dynamics, measurement geometry, and process noise intensity. These topics should be considered in more detailed investigations and are being studied at this time.

The results in Refs. 1-3 were generated on a large-wordlength machine. As such, truncation errors due to numerical integration will predominate over roundoff errors. Further evaluations of the algorithms on computers with smaller word sizes are warranted. However, even on a large-wordlength machine, the effect of integration error can be an important determinant in algorithm selection.[2] The results presented in Ref. 2 give such an indication but are by no means exhaustive.

Bierman's questions regarding the validity of the numerical results presented in Ref. 1 do not focus on the complexity of this question. Based on the extensive numerical simulations reported in Refs. 1-3, it is evident that the "along-track" error in the GPS-navigation filter considered in these investigations involves a complicated interaction of three error sources. Numerical integration error, clock bias, and drag coefficient error all lead to "along-track" error. The coupling of these error sources, while not understood at this time, is almost surely the cause of the numerical integrator dependence reported.

Finally, the study cited by Bierman which was carried out for the GPS phase I evaluation of the Tapley-Choe algorithm[6] was, to the authors' knowledge, not exhaustive. Moreover, the application involved a backward mapping of the observations (not a forward mapping of the state) over time periods of up to 24 hours, assuming the presence of process noise in the state equation. $B_D Q_D B_D^T$, must be time dependent, and it must be evaluated correctly or the algorithm will surely fail in this application. The results will depend on the value of the continuous process noise model used to evaluate the integral. After the observation mapping was accomplished, the algorithm proposed in Ref. 6 was applied to a "static" system where a recursive "update" of the state at the initial epoch was attempted. Since the results referred to by Bierman are not available to the authors, more specific comments cannot be offered. However, based on the previous statements, it is safe to say that the experience in this application is not particularly relevant to the application discussed in Ref. 1.

Concluding Remarks

The intent of the presentation of the results in Ref. 1 is not to denigrate the performance of the discrete *U-D* factorization time update algorithm. It has obviously proven itself to be an efficient, accurate, and stable method in a number of applications.

The continuous *U-D* update form is offered as a potential alternative to the discrete form. The results in Ref. 1 show evidence that, in some instances, the continuous update is competitive or even faster than the discrete form. It also offers some options for optimizing storage requirements. Clearly, these results are illustrative and not exhaustive. As suggested by Bierman, additional study is required to test the stability and adaptability of the algorithm.

References

[1] Tapley, B.D. and Peters, J.G., "Sequential Estimation Algorithm Using a Continuous UDU^T Covariance Factorization," *Journal of Guidance and Control,* Vol. 3, July-August 1980, pp. 326-331.

[2] Tapley, B.D., Peters, J.G., and Schutz, B.E., "Relative Performance of Algorithms for Autonomous Satellite Orbit Determination," *Journal of the Astronautical Sciences* (in press).

[3] 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 of Orbital Mechanics, Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, Texas, Report No. TR 79-1, March 1980.

[4] Bierman, G.J., "*U-D* Filter Time Propagation," Jet Propulsion Laboratory, Interoffice Memorandum 314.4-40, April 12, 1977.

[5] Jazwinski, A.H., *Stochastic Processes and Filtering Theory,* Academic Press, New York, 1970, pp. 84-85.

[6] 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.