define the dynamic error E as

$$E=\int_0^{t_f}\left\{\sum_{i=1}^{K}[f_i(X,V,t)-\dot{X}_i(t)]^2\right\}dt \tag{5}$$

The optimal control problem now reduces to minimizing J given by Eq. (2) under the constraint

$$E=0 \tag{6}$$

in order to obtain a_{im} and b_{jm} ($i=1,2,...,K$; $j=1,2,...,L$; $m=1,2,...,M$). This can be achieved by a multivariable search method such as SWIFT.[1] In the problem of minimum time $g=1$ and $J=t_f$.

The first step in the iterative procedure is to choose M, the number of functions to be assumed for an adequate expansion in Eqs. (3) and (4), and N, the number of discrete points $t_0,...,t_f=(t_{N-1})$ in the interval $(0,t_f)$. Each time interval is given by $\Delta t=t_{i+1}-t_i$ so that the total time $t_f=N\Delta t$. Choosing Δt, a_{im}, b_{jm} as the parameters for optimization, one can use Eqs. (3-6) repeatedly before a preset convergence criterion is satisfied.

Example

The above procedure is applied to obtain the thrust angle history for transferring a space probe from Earth's orbit to Mars' orbit. The dynamic equations of a space vehicle are given in Ref. 2 and illustrated in Fig. 1. The nondimensional equations can be shown to be

$$\dot{X}_1=X_2 \qquad \dot{X}_2=\frac{X_3^2}{X_1}-\frac{1}{X_1^2}+\frac{T\sin\theta}{m}$$

$$\dot{X}_3=-\frac{X_2X_3}{X_1}+\frac{T\cos\theta}{m} \tag{7}$$

where $m=1-\dot{m}t$ and where mass and length are measured by the initial values m_0, r_0 and time is measured by $t_0=r_0/V_0$, where V_0 is the initial velocity. θ is measured in radians; thrust is measured by $T_0=m_0r_0/t_0^2$.

The nondimensional burning rate and thrust are assumed constant such that $\dot{m}=0.0749$ and $T=0.1405$. These values are appropriate for a 35 kW propulsion system. The weight at escape is 1500 lb, specific impulse is 5700 s, and rate of burning is 1.95 lb/day. These are very close to the parametric values of the vehicle to be used with the Saturn rocket and SNAP-8 reactor.[4] Boundary conditions turn out to be: $X_1(0)=1.0$, $X_2(0)=0.0$, $X_3(0)=1.0$; $X_1(t_f)=1.525$, $X_2(t_f)=0$ and $X_3(t_f)=0.8098$. In this problem θ is the control variable.

Table 1 gives the minimum time t_f, the dynamic error E for various values of N and M.

We note that the first solution which gives $t_f=5.83$ is the best solution as the dynamic error constraint ($E=8.1\times10^{-8}$) is adequately satisfied. The time taken for the space probe is 338 days ($=5.83\times365/2\pi$). This value compares extremely well with that of Viking-2 which took off on Sept. 9, 1975 and swung into Mars' orbit on Aug. 7, 1976 (about 333 days). Figure 2 gives the state and control time histories. To the authors' knowledge a solution for this particular problem with conventional methods of solving the TPBVP is not available for comparison. Taylor et al.[2] give the minimum time as 3.39 with $E=10^{-4}$ while Williamson[3] gives $t_f=3.4199$. These values and also the other cases of Table 1 do not give realistic time.

References

[1]Sheela, B.V. and Ramamoorthy, P., "SWIFT—A New Constrained Optimization Technique," *Computer Methods in Applied Mechanics and Engineering,* Vol. 6, Dec. 1975, p. 309.

[2]Taylor, J.M. and Constantinides, C.T., "Optimization: Application of the Epsilon Method," *IEEE Transactions on Automatic Control,* Vol. AC 17, Feb. 1972, p. 128.

[3]Williamson, W.E., "Use of Polynomial Approximations to Calculate Suboptimal Controls," *AIAA Journal,* Vol. 9, Nov. 1971, p. 2271.

Technical Comments

AIAA 81-4172

Comment on "Sequential Estimation Algorithm Using a Continuous UDU^T Covariance Factorization"

Gerald J. Bierman*
Factorized Estimation Applications, Inc., Canoga Park, Calif.

TAPLEY and Peters[1] propose replacing the discrete-time *U-D* covariance factor Gram-Schmidt time update described in Refs. 2 and 3. Their approach is to modify an idea due to Andrews,[4] and they present a set of *U-D* element differential equations. To justify their results, they consider a LANDSAT-D satellite navigation application. Their comparison of results and the conclusions they draw do not, however, adequately describe the relative merits of differential equation *U-D* element propagation and discrete time propagation. Our purpose is to briefly highlight key comparison items that were not discussed in Ref. 1, and to clarify some misleading statements made therein.

Received Sept. 29, 1980.

*President.

Problem Introduction

To best understand the distinction between the continuous and discrete-time algorithm we consider the linear continuous time model

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

where A and B_c may be time dependent, $\xi(t)$ is white process noise with diagonal intensity $Q_c=Q_c(t)$, and the symbol $N(\bar{x}_0,P_0)$ indicates that $x(t_0)$ is normal, with mean $\bar{x}_0$ and covariance P_0. For simplicity of exposition the usual assumption made is that $\xi(t)$ is independent of $x(t_0)$. Because Eq. (1) is linear, one can express the solution to the differential equation as

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

where $\{B_Dw_j\}$ is a white noise sequence and $\Phi(t_{j+1},t_j)$ is the solution to the linear differential equation

$$\frac{d}{dt}\Phi(t,t_j)=A\Phi(t,t_j) \qquad (\Phi(t,t_j)=I \quad \text{for} \quad t=t_j) \tag{3}$$

$$B_D w_j = \int_{t_j}^{t_{j+1}} \Phi(t_{j+1},s) B_c(s) \xi(s) \mathrm{d}s \tag{4}$$

$$B_D Q_D B_D^T = \int_{t_j}^{t_{j+1}} \Phi(t_{j+1},s) B_c(s) Q_c(t) B_c(s)^T \Phi^T(t_{j+1},s) \mathrm{d}s \tag{5}$$

The point to emphasize here is that the continuous and discrete models [Eqs. (1) and (2)] are equivalent; there is no approximation involved. Algebraically equivalent estimates of $x(t_{j+1})$, $\hat{x}(t_{j+1}|t_j)$ can be obtained from estimates of $x(t_j), \hat{x}(t_j|t_j)$, by either of the following methods:

$$\dot{x} = Ax \qquad x(t_j) = \hat{x}(t_j|t_j) \tag{1'}$$

$$\hat{x}(t_{j+1}|t_j) = \Phi(t_{j+1},t_j)\hat{x}(t_j|t_j) \tag{1''}$$

where the differential equation solution at t_{j+1} is $\hat{x}(t_{j+1}|t_j)$. The corresponding continuous and discrete estimate error prediction covariance relations are

$$\dot{\Sigma} = A\Sigma + (A\Sigma)^T + Q_c \qquad \Sigma(t_j) = \hat{P}_j(t_j|t_j) \tag{6}$$

$$\hat{P}(t_{j+1}|t_j) = \Phi\hat{P}(t_j|t_j)\Phi^T + B_D Q_D B_D^T \tag{7}$$

where the differential equation solution at t_{j+1}, $\Sigma(t_{j+1})$ is $\hat{P}(t_{j+1}|t_j)$ and $\Phi = \Phi(t_{j+1},t_j)$.

Tapley and Peters,[1] propose modifying Eq. (6) and produce differential equations for U and D, where $P = UDU^T$. In the discrete-time formulation,[5] we employ numerically reliable matrix modification algorithms,[3] to carry out the propagation. Tapley and Peters claim that they get more accurate results when they integrate their continuous formulas numerically than when the discrete-time Gram-Schmidt algorithm is applied. We discuss this part a bit further on, but note here that the numerical results that they report seem to be attributable to the effects of Euler integration error on transition matrix integration as opposed to $\dot{U}$-$\dot{D}$ integration.

Comparison of Tapley-Peters *U-D* Differential Equation Propagation and Discrete-Time *U-D* Propagation

With this capsulated background we are in position to compare and discuss the continuous and discrete-time approaches. The Tapley-Peters *U-D* differential equations are based on the covariance result [Eq. (6)], and their result is to solve

$$\dot{U}D + U\dot{D}/2 = AUD + (Q/2 + C)U^{-T} \tag{8}$$

where C, skew symmetric, is chosen so that $\dot{U}$ will be triangular. The discussion in Ref. 1 exaggerates the benefits of the *U-D* differential equation approach in the areas of storage, efficiency, and accuracy.

Storage

It is stated that the *U-D* differential equation method requires less storage than does the discrete model approach because, instead of integrating a matrix differential equation for Φ and propagating U and D discretely, we need only propagate $\dot{U}$ and $\dot{D}$. The nature of Eq. (8) requires, according to Tapley and Peters, that we have matrices M, T, and C in addition to $\dot{U}$-$\dot{D}$ and *U-D*. Having a smaller number of more complicated equations does not necessarily reduce storage. An important point only briefly touched on in their satellite example is that the transition matrix can, in many important applications, be solved analytically (thus reducing computational effort). Moreover, transition matrices often have many known zero and constant entries. On the other hand, the $\dot{U}$-$\dot{D}$ equations do not generally have sparse structure even when Φ does. Incidentally, discrete-time *U-D* time propagation usually involves considerably less storage than is implied in the description given in Ref. 1. Thus, one should not abandon the discrete model in favor of the *U-D* differential equation approach based on storage considerations.

Efficiency

Claims of efficiency for the *U-D* differential equation approach are misleading. Efficiency is a function of both the algorithm and the implementation. To illustrate the significance of implementation, we note that the measurement update results of Table 2 in Ref. 1 indicate that the *U-D* measurement update is about 13% more expensive than the optimal Kalman filter update. With efficient code implementation, our experience (e.g., see Table VI of Ref. 6) shows that the computational difference should be less than 2%, and for the satellite example involved in this case an efficient *U-D* mechanization actually turns out to be more efficient. We note in passing that inattention to details of implementation can more than double computation and storage requirements. We draw attention to the remark of Tapley and Peters that "... square root filter operations often incur computation time penalties as the price for increased numerical stability." It has been this author's experience in almost every application either that the computational costs and financial cost differences are inconsequential or that by careful implementation one can arrange the square root filter programs so that their run times are within 10-15% of the time required for the corresponding conventional Kalman filter program. It is not widely known, but for many applications the triangular structure of the square root algorithms leads to implementations that are more efficient than are their covariance counterparts. The important point is that when efficiency is paramount, one must maximally exploit triangular structure, sparseness, code compactness (i.e., eliminate unnecessary "do loops," index calculations, etc.), and unnecessary input/output.

Attention turns now to the question of algorithm efficiency. Important features of the discrete-time approach that are not shared by the *U-D* differential equation approach include the following:

1) Analytic transition matrices, either exact closed-form solutions or else sufficiently accurate approximations, are often used. On the other hand, the nonlinearity of the *U-D* equations, [Eq. (8)] rules out analytic propagation (even for simple examples).

2) Time invariance of the dynamics, when applicable, allows one to avoid calculating transition matrices at each step. The *U-D* integration is not simplified by the time invariance of A.

3) Subsystem dynamic decoupling, when applicable, permits one to compute transition matrices of the component subsystems. On the other hand, *U-D* integration must include integration of all statistically cross coupled terms.

4) A key feature of the transition matrix approach is that the same dynamic model is often used for a sequence of covariance analyses. For such problems one can use stored transition matrices. Since in complex problems transition matrix integration can be a major cost of the filter process, the resulting savings can be impressive. This is, for example, frequently the state of affairs in spacecraft navigation accuracy analyses.

5) When the noise states are simple Markov colored noise, or integrals of such, one can employ efficient rank-1 matrix update techniques, as is discussed in Ref. 3. The *U-D* differential equations, as given in Ref. 1, do not directly benefit from such structure.

6) Integration step size for the transition matrix computation is determined by the dynamics. Integration of the $\dot{U}$-$\dot{D}$ equations is determined in addition by the a priori estimate statistics, the dynamic process noise intensities, the data geometry (observation coefficients), and the data quality (measurement variances). One could ignore these facts in selecting *U-D* differential equation step size, but the effect on accuracy could be disastrous.

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.