Square-Root State Estimation for Second-Order Large Space Structures Models

Yaakov Oshman* and Daniel J. Inman†
State University of New York at Buffalo, Buffalo, New York
and
Alan J. Laub‡
University of California, Santa Barbara, Santa Barbara, California

Two square-root filtering algorithms are developed for large space structures that are modeled by second-order, continuous-time, finite, dynamic models. The first filter, which assumes a continuous-time measurement system, is a single-stage continuous algorithm that is based on the V-Lambda square-root method for the solution of a generalized Riccati equation. The second measurement system considered is of a discrete-time type, for which the resulting estimator is a hybrid continuous/discrete one. Both estimators are based on the spectral decomposition of the estimation error covariance matrix. Thus, they continuously provide the user with the covariance spectral factors. This distinct feature of the V-Lambda algorithms is valuable in ill-conditioned cases, in which an insight into the estimation process is needed to reveal singularities and to identify state subsets that become nearly dependent. Moreover, using the orthogonality property of the covariance eigenvectors, an orthogonalization step is added to the algorithms to enhance their accuracy in cases where simple, unsophisticated software is to be used. Two different methods for performing the orthogonalization are suggested. A typical filtering example is used to demonstrate the square-root nature of the new filters.

I. Introduction

THE application of modern multivariable control theory to the design of advanced large space structures (LSS) has received a great deal of attention in the last decade.[1-3] In most cases, the system equations have a linear, vector, second-order form, which usually results from some finite approximation to the distributed parameter description by partial differential equations. Modern linear system theory, however, requires that the mathematical model of the system be formulated in a vector first-order form. A direct application of modern control theory results to LSS problems, therefore, makes it necessary to transform the original second-order formulation to an equivalent vector first-order form. Several ways of performing this transformation exist; however, they all share the common disadvantage in that they prohibit the exploitation of any special properties that the system matrices [M, D, or K in Eq. (1a)] may have (i.e., symmetry, definiteness, and/or sparsity). To make the last point clearer, consider the following continuous-time, stochastic equation, which mathematically models LSS:

$$M\ddot{x} + D\dot{x} + Kx = Bu + Gw \tag{1a}$$

where $x(t) \in R^n$ is the system's generalized position vector; $u(t) \in R^q$ is the control input; $w(t) \in R^p$ is a zero-mean, white, Gaussian noise process with covariance $E[w(t)w(s)^T] = Q(t)\delta(t-s)$; M is the symmetric, positive definite mass matrix; K is the symmetric, positive semidefinite stiffness matrix; and D is the symmetric damping matrix.

It is assumed that only a linear combination of some subsets of the system's position and velocity components is measured:

$$y = H_1 x + H_2 \dot{x} + v \tag{1b}$$

where $y \in R^m$ is the measurement vector; and $v \in R^m$ is a zero-mean, white, Gaussian measurement noise process with intensity $R(t)$, which is uncorrelated with $w(t)$ (the extension to the correlated case being straightforward).[4]

It is also assumed that the initial generalized position and velocity of the structure are

$$x(0) = x_0 \qquad \text{and} \qquad \dot{x}(0) = \dot{x}_0 \tag{1c}$$

where x_0 and $\dot{x}_0$ are jointly Gaussian with known means and second moments and have no correlation with the noise processes $w(t)$ and $v(t)$.

Now, by augmenting the state vector, the "classical" equivalent first-order realization of Eqs. (1a-1c) is

$$\frac{d}{dt}\begin{pmatrix} x \\ \dot{x} \end{pmatrix} = \begin{pmatrix} 0 & I \\ -M^{-1}K & -M^{-1}D \end{pmatrix}\begin{pmatrix} x \\ \dot{x} \end{pmatrix} + \begin{pmatrix} 0 \\ M^{-1}B \end{pmatrix}u + \begin{pmatrix} 0 \\ M^{-1}G \end{pmatrix}w \tag{2a}$$

$$y = [H_1 \quad H_2]\begin{pmatrix} x \\ \dot{x} \end{pmatrix} + v \tag{2b}$$

$$\begin{pmatrix} x(0) \\ \dot{x}(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ \dot{x}_0 \end{pmatrix} \tag{2c}$$

However, by examining the realization (2), it is clear that any symmetry, definiteness, and/or sparsity of the system matrices is destroyed and cannot be used effectively in the algorithm implementation. In this respect it should be noted that it is very common that the system matrices (all or some) are sparse, in addition to their being symmetric, because of their compu-

Based in part on AIAA paper presented at the 6th VPI&SU/AIAA Symposium on Dynamics and Control of Large Structures, Blacksburg, VA, June 29-July 1, 1987; and on AIAA Paper 87-2389 presented at the AIAA Guidance, Navigation, and Control Conference, Monterey, CA, Aug. 17-19, 1987; received Feb. 5, 1988; revision received June 16, 1988.

*Visiting Assistant Professor, Department of Mechanical and Aerospace Engineering; currently, Lecturer, Faculty of Aerospace Engineering, Technion – Israel Institute of Technology, Haifa, Israel.

†Professor, Department of Mechanical and Aerospace Engineering. Member AIAA.

‡Professor, Department of Electrical and Computer Engineering.

tation by efficient, finite-element programs that arrange the node ordering in the structural model to achieve this sparsity. Sparse matrix technology has been developed very rapidly in recent years, and highly efficient algorithms exist for the solution of various problems in which sparse matrices are involved.[5,6] The transformation of the second-order model [Eq. (1)] to the first-order realization [Eq. (2)] prohibits the use of these sparse-matrix algorithms and thus may increase the computational burden considerably.

For this reason, as well as others, several works that have appeared recently address the analysis of second-order models directly, without first transforming them to the standard first-order form.[7-11] Of special interest is the recent work of Hashemipour and Laub,[11] which motivated the research presented in this paper. They addressed the problem of state estimation in second-order systems by developing Kalman filtering (KF) equations for continuous-time, second-order systems (continuous dynamics and continuous measurement) and for discrete-time, second-order systems (discrete dynamics and discrete measurement) that are obtained from the original continuous systems by some type of approximation. Their equations retain the original structure possessed by the system parameters, thus enabling the exploitation of any special structure of these parameters in the implementation of the filter. As in the conventional Kalman filter algorithm, in which a matrix differential equation has to be solved for the estimation error covariance matrix (the Riccati equation) in the continuous case, or a matrix recursive equation in the discrete case, the results presented in Ref. 11 are based on the propagation of covariance matrices.

It is now widely recognized that the filtering algorithms presented by Kalman[12] and Kalman and Bucy[13] may suffer from numerical instability. Soon after the introduction of these algorithms, it was shown that their implementation in practice, especially on short-word-length computers, may lead to the computation of negative-definite covariance matrices,[14-16] which, in turn, may cause filter divergence. Square-root (SR) filtering algorithms were developed to overcome these difficulties. These algorithms use a decomposition of the error covariance matrix into its SR factors, which replace the covariance matrix in each stage of the computation. In this way the covariance matrix itself is never explicitly computed. This ensures that the (implicit) covariance matrix is always symmetric non-negative definite, stabilizes the filtering algorithm, and greatly enhances its accuracy in ill-conditioned cases.[17] Since the decomposition of a nonnegative matrix into its SR factors is not unique, several SR methods have been developed that are based on different SR decompositions. Among these are those based on Q-R factorizations;[18] Bierman's *U-D* method,[19,20] which uses a UDU^T decomposition of the covariance where U is a unit upper triangular matrix and D is diagonal; and the recently introduced V-lambda algorithms,[16,21] which use the spectral decomposition of the covariance into a $V\Lambda V^T$ form where V is the matrix whose columns are the eigenvectors of the covariance and Λ is the diagonal eigenvalue matrix.

As pointed out in Ref. 11, actual implementation of the second-order (covariance) estimation algorithms should be performed in factorized or SR form because of reasons similar to those for using SR algorithms in the first-order (conventional KF) case. The purpose of this paper is to apply the notion of SR filtering to LSS problems by presenting SR solutions to the problem of state estimation in linear vector second-order systems. Two algorithms are presented here: one for the continuous-time, second-order model (continuous dynamics and continuous measurement) and a hybrid algorithm for the (more practical) case of continuous dynamical model augmented by a discrete-time measurement. Both algorithms are based on the aforementioned spectral decomposition of the covariance matrix and as such belong to the V-lambda class of SR filters, thus inheriting their excellent numerical characteristics.[16,21-23] Moreover, the fact that the algorithms are formulated in terms of the eigenvectors and eigenvalues (eigenfactors) of the covariance matrix renders them especially important in cases where continuous monitoring of the eigenfactors is needed to reveal singularities as they occur and to identify those state subsets that are nearly dependent.[17,18,24]

In the following section, the continuous-time algorithm is developed. This algorithm is modified in Sec. III to incorporate a discrete-time measurement, thus forming a hybrid continuous/discrete V-Lambda filter. In Sec. IV the orthogonality property of the covariance eigenvectors is exploited, using two different approaches (Baruch and Bar-Itzhack's iterative orthogonalization[25] and a method based on the singular value decomposition[26]) to enhance the filter accuracy by reducing the numerical integration error. A filtering example is presented in Sec. V, which serves to demonstrate the numerical robustness of the new filters. The paper is concluded in Sec. VI.

II. Continuous-Time Square Root Algorithm

In this section we develop a SR filtering algorithm for systems modeled by second-order, continuous-time models. To simplify the ensuing development, we shall assume, without loss of generality, that the only input to the system is the stochastic process noise, the inclusion of the deterministic control input effects being straightforward. The flexible structure we shall deal with is mathematically modeled by Eqs. (1), which, following Ref. 7, will be written in an implicit form as

$$\begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix}\begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{pmatrix} 0 & I \\ -K & -D \end{pmatrix}\begin{pmatrix} x \\ \dot{x} \end{pmatrix} + \begin{pmatrix} 0 \\ G \end{pmatrix} w \tag{3a}$$

$$y = [H_1 \quad H_2]\begin{pmatrix} x \\ \dot{x} \end{pmatrix} + v \tag{3b}$$

This can be written more compactly as

$$E\dot{q} = Fq + \bar{G}w \tag{4a}$$

$$y = \bar{H}q + v \tag{4b}$$

with obvious definitions of the augmented state vector q and the system matrices E, F, $\bar{G}$, and $\bar{H}$. Also, we define the estimation error covariance matrix by

$$P = E\{[q - \hat{q}][q - \hat{q}]^T\} \tag{5}$$

where $\hat{q}$ is the estimated augmented state, and $E\{\bullet\}$ stands for the expectation operator.

The KF Riccati equation for the estimation error covariance $P(t)$ is given by the generalized Riccati equation:[27,28]

$$E\dot{P}E^T = FPE^T + EPF^T + \bar{G}Q\bar{G}^T - EP\bar{H}^TR^{-1}\bar{H}PE^T \tag{6}$$

As is well known, this equation forms the core of the continuous-time KF algorithm for state estimation in systems describable by models such as Eq. (1). To develop the algebraically equivalent (but numerically superior) SR algorithm, we need a result from an earlier paper (Ref. 16), which relates the rate of change of the eigenfactors of a self-adjoint matrix to the rate of change of the matrix and is restated here for convenience.

Theorem 1. Rate of change of the eigenfactors of a self-adjoint matrix that depends on a real parameter. Let $P(t)$ be an $n \times n$ complex matrix function, that depends on the real parameter t, and let $P(t)$ satisfy the following two conditions for every $t \in R$:

1) $P(t)$ is self-adjoint, i.e., $P(t) = P(t)^*$, where the asterisk denotes the conjugate transpose matrix.

2) $P(t)$ is an analytic function of the real variable t. Then, there exist scalar functions $\{\lambda_i(t)\}_{i=1}^n$ and a matrix-valued function $V(t)$, which are analytic for $t \in R$ and possess the following properties for every $t \in R$:

a) $P(t) = V(t)\ \mathrm{diag}\{\lambda_1(t), \lambda_2(t), \ldots, \lambda_n(t)\}V(t)^{-1}$; b) $V(t)^* V(t) = I$, where I is the identity matrix. Furthermore, defining

the diagonal matrix Λ to be

$$\Lambda = \text{diag}\{\lambda_1,\lambda_2,\ldots,\lambda_n\}$$

and the ith column of $V(t)$ to be $v_i(t)$, we have

$$\frac{d\lambda_i(t)}{dt} = v_i(t)^* \frac{dP(t)}{dt} v_i(t), \qquad i = 1,2,\ldots,n \tag{7a}$$

and

$$\dot{v}_i(t) = \sum_{\substack{q=1 \\ \lambda_i \neq \lambda_q}}^{n} \frac{v_q(t)^* \frac{dP(t)}{dt} v_i(t)}{\lambda_i(t) - \lambda_q(t)} v_q(t), \qquad i = 1,2,\ldots,n \tag{7b}$$

Since $P(t)$, the solution matrix of the Riccati equation, satisfies the conditions of the theorem, the results stated in the theorem can be applied in our case. Identifying the functions $\{\lambda_i(t)\}_{i=1}^n$ with the eigenvalues of $P(t)$ and the columns $v_i(t)$ of $V(t)$ with its corresponding eigenvectors, we note that the main theorem relates, by way of Eq. (7), the rate of change of the eigenfactors of $P(t)$ to the rate of change of $P(t)$ itself. To obtain the required SR algorithm, however, we need to develop differential equations for the eigenfactors, in which no explicit computation of the covariance will appear; that is, we need to develop expressions of the following form:

$$\dot{\lambda}_i(t) = f_i[\{\lambda_j(t)\}_{j=1}^n, \{v_j(t)\}_{j=1}^n], \qquad i = 1,2,\ldots,n \tag{8a}$$

$$\dot{v}_i(t) = g_i[\{\lambda_j(t)\}_{j=1}^n, \{v_j(t)\}_{j=1}^n], \qquad i = 1,2,\ldots,n \tag{8b}$$

where f_i, g_i are functions of the covariance eigenfactors and, possibly, the system parameter matrices. Noting Eq. (7) of Theorem 1, we see that to cast these equations in the required form [Eq. (8)] we have to express the numerators in the right side of Eq. (7b) as functions of the eigenfactors. This will be done next.

We start by rewriting Eq. (6) as

$$\dot{P} = E^{-1}FP + PF^TE^{-T} + E^{-1}\bar{G}Q\bar{G}^TE^{-T} - P\bar{H}^TR^{-1}\bar{H}P \tag{9}$$

We note here that, at least in theory, we may do so, since the mass matrix M (and hence also the matrix E) is assumed to be invertible; however, because the potentially problematic nature of the mass matrix inversion in practice is recognized, this inversion will be circumvented at a later stage of the derivation. Casting Eq. (6) in the form of Eq. (9) may therefore be considered a purely didactic step. Premultiplying Eq. (9) by v_q^T and postmultiplying it by v_i, we have

$$v_q^T\dot{P}v_i = v_q^TE^{-1}FPv_i + v_q^TPF^TE^{-T}v_i + v_q^TE^{-1}\bar{G}Q\bar{G}^TE^{-T}v_i - v_q^TP\bar{H}^TR^{-1}\bar{H}Pv_i \tag{10}$$

Using the relations that must be satisfied by the eigenvalue-eigenvector pairs

$$Pv_i = \lambda_i v_i$$

and

$$v_q^TP = \lambda_q v_q^T$$

Equation (10) takes the form

$$v_q^T\dot{P}v_i = \lambda_i v_q^TE^{-1}Fv_i + \lambda_q v_q^TF^TE^{-T}v_i + v_q^TE^{-1}\bar{G}Q\bar{G}^TE^{-T}v_i - \lambda_i\lambda_q v_q^T\bar{H}^TR^{-1}\bar{H}v_i \tag{11}$$

in which no explicit usage of the covariance matrix is made. However, as noted before, the last expression does contain an inverse of the matrix E, which implies an inversion of the mass matrix. To avoid the numerical difficulties that may be associated with this inversion being ill-conditioned or in near-singular cases, we use the following transformation, by which the set of eigenvectors of P is transformed into another set of "weighted eigenvectors":

$$z_i := E^{-T}v_i \tag{12}$$

from which we clearly have

$$v_i^TE^{-1} = z_i^T \tag{13}$$

$$v_i = E^Tz_i \tag{14}$$

and

$$v_i^T = z_i^TE \tag{15}$$

Substituting Eqs. (12-15) into Eq. (11) yields

$$\begin{aligned} v_q^T\dot{P}v_i &= \lambda_i z_q^TFE^Tz_i + \lambda_q z_q^TEF^Tz_i + z_q^T\bar{G}Q\bar{G}^Tz_i \\ &\quad - \lambda_i\lambda_q z_q^TE\bar{H}^TR^{-1}\bar{H}E^Tz_i \\ &= z_q^T\{\lambda_iFE^T + \lambda_qEF^T + \bar{G}Q\bar{G}^T \\ &\quad - \lambda_i\lambda_qE\bar{H}^TR^{-1}\bar{H}E^T\}z_i \end{aligned} \tag{16}$$

Now define the set of scalar functions $\{\gamma_{iq}\}_{i,q=1}^n$ by

$$\gamma_{iq} = z_q^T\{\lambda_iFE^T + \lambda_qEF^T + \bar{G}Q\bar{G}^T - \lambda_i\lambda_qE\bar{H}^TR^{-1}\bar{H}E^T\}z_i, \qquad i,q = 1,2,\ldots,n \tag{17}$$

Substituting the matrices E, F, $\bar{G}$, and $\bar{H}$ of Eq. (4) into Eq. (17), γ_{iq} can be rewritten as

$$\gamma_{iq} = z_q^TT_{iq}z_i \tag{18}$$

where T_{iq} is a matrix-valued function defined by

$$T_{iq} := \begin{bmatrix} -\lambda_i\lambda_qH_1^TR^{-1}H_1 & (\lambda_iM - \lambda_qK)^T - \lambda_i\lambda_qH_1^TR^{-1}H_2M^T \\ \lambda_qM - \lambda_iK - \lambda_i\lambda_qMH_2^TR^{-1}H_1 & -(\lambda_iDM^T + \lambda_qMD^T) + GQG^T - \lambda_i\lambda_qMH_2^TR^{-1}H_2M^T \end{bmatrix} \tag{19}$$

Employing the definition of γ_{iq} in Eq. (16) yields

$$v_q^T\dot{P}v_i = \gamma_{iq} \tag{20}$$

Returning now to the equations for the rate of change of the eigenvectors [Eq. (7b)] and using Eq. (20), we have

$$\dot{v}_i = \sum_{\substack{q=1 \\ \lambda_i \neq \lambda_q}}^{n} \frac{\gamma_{iq}}{\lambda_i - \lambda_q} v_q, \qquad i = 1,2,\ldots,n \tag{21}$$

Premultiplying Eq. (21) by E^{-T} and noting the definition in Eq. (12), we obtain

$$\dot{z}_i = \sum_{\substack{q=1 \\ \lambda_i \neq \lambda_q}}^{n} \frac{\gamma_{iq}}{\lambda_i - \lambda_q} z_q, \qquad i = 1,2,\ldots,n \tag{22}$$

Also, from Eqs. (7a) and (20) we have

$$\dot{\lambda}_i = \gamma_{ii}, \qquad i = 1,2,\ldots,n \tag{23}$$

Equations (22) and (23) together form the required set of coupled differential equations for the variables $\{\lambda_i,z_i\}_{i=1}^n$. To cast these equations in matrix form, we introduce the follow-

ing definitions. Let Z be the matrix whose columns are the weighted eigenvectors $\{z_i\}_{i=1}^n$, and let Ω be the skew-symmetric matrix whose element in the qth row and ith column is

$$\Omega_{qi} = \begin{pmatrix} \gamma_{iq}/(\lambda_i - \lambda_q) & \text{for } \lambda_i \neq \lambda_q \\ 0 & \text{for } \lambda_i = \lambda_q \end{pmatrix} \tag{24}$$

Also let Γ be the diagonal matrix defined by

$$\Gamma := \text{diag}\{\gamma_{11}, \gamma_{22}, \ldots, \gamma_{nn}\} \tag{25}$$

With these definitions, Eqs. (22) and (23) are rewritten in matrix form as

$$\dot{Z} = Z\Omega \tag{26a}$$

$$\dot{\Lambda} = \Gamma \tag{26b}$$

The initial conditions for the integration are

$$Z(t_0) = Z_0 \tag{27a}$$

$$\Lambda(t_0) = \Lambda_0 \tag{27b}$$

where Z_0 is computed at the beginning of the integration process by

$$Z_0 = E^{-T} V_0 \tag{28}$$

and Λ_0 and V_0 are obtained by a spectral decomposition of the initial error covariance

$$P_0 = V_0 \Lambda_0 V_0^T \tag{29}$$

Remarks.

1) Note that the inverted mass matrix is not required in the numerical integration process itself. This matrix is required only once, when the initial conditions are computed [Eq. (28)]. Thus, in cases when the mass matrix inversion is troublesome, it can be carried out using high-precision software, without affecting the complexity of the whole filtering scheme.

2) The estimation error covariance matrix is not needed and is therefore not computed at any stage of the process. However, if so desired, it can be readily reconstructed by

$$P = E^T Z \Lambda Z^T E \tag{30}$$

3) Examination of Eq. (24) reveals that, if some of the eigenvalues become very closely spaced, the computation might run into difficulties because the magnitudes of the derivatives of the corresponding eigenvectors may become too large. A possible solution to this problem is the use of a special routine that can handle stiff differential equations. However, the use of such special programs may sometimes be inefficient or even impossible. In these cases, another solution to the stiffness problem is needed, and such a solution is given in Ref. 16. By thoroughly treating the case of clustering eigenvalues, it was shown there that the potential numerical problem that may arise in such cases can be easily circumvented without suffering any practical reduction in accuracy by implementing the following modified version of Eq. (24):

$$\Omega_{qi} = \begin{pmatrix} \dfrac{\gamma_{iq}}{\lambda_i - \lambda_q} & \text{for } \left|\dfrac{\gamma_{iq}}{\lambda_i - \lambda_q}\right| < \Omega_{\max} \\ 0 & \text{for } \lambda_i = \lambda_q \\ \Omega_{\max}\, \text{sign}\left\{\dfrac{\gamma_{iq}}{\lambda_i - \lambda_q}\right\} & \text{for } \left|\dfrac{\gamma_{iq}}{\lambda_i - \lambda_q}\right| \geq \Omega_{\max} \end{pmatrix} \tag{31}$$

where $\Omega_{\max}$ is determined before computation, in accordance with the capability of the software used and the accuracy sought.

Having developed the propagation equations for the covariance spectral factors, the equations for the state estimate are addressed next. The KF equation for the system (4a) with measurement (4b) is

$$E\dot{\hat{q}} = F\hat{q} + G_K\tilde{y} \tag{32}$$

where the innovations process $\tilde{y}$ is defined as

$$\tilde{y} := y - \bar{H}\hat{q}$$

and G_K is the Kalman gain matrix, which can be computed by

$$G_K = V\Lambda V^T \bar{H}^T R^{-1} \tag{33}$$

The presentation of the SR filter for the continuous-time model is thus complete. In the next section we address the case of a system that is modeled by a continuous-time dynamics equation and a discrete-time measurement (the continuous/discrete case).

III. Continuous/Discrete Square Root Filter

The continuous-time filter introduced in the preceding section is based on the assumption that the measurements are acquired in a continuous fashion. Although there are cases in which the measurements are taken at a very high rate (thus approximating continuous-time measurements for practical purposes), it must still be admitted that this assumption is not a very realistic one. This is so because in most real-life applications continuous measurements are either unnecessary or cannot be processed in real time for reasons of limited computational power (especially when on-board computers are used that are not fully dedicated to the estimation task.) Thus, although the continuous-time filter has its theoretical value, there is a need to develop an estimation algorithm that will accommodate discrete-time measurements. This algorithm will also be more in tune with digital computers, which are heavily used in today's applications. This section is concerned with the development of such a filtering algorithm.

Although it will be assumed that the measurements are acquired at discrete-time epochs, a continuous-time dynamics model will still be used. This is in agreement with many applications (e.g., large flexible structures) in which the "exact" mathematical model of the system is formulated naturally in continuous-time domain; in these applications a passage to discrete-time models would necessitate further approximations (in addition to those already used in the derivation of the "exact," continuous model). Thus, the mathematical model chosen here consists of the dynamics Eq. (1a), rewritten here for completeness (again, it is assumed that there is no deterministic input):

$$M\ddot{x} + D\dot{x} + Kx = Gw \tag{34a}$$

augmented with the following discrete-time measurement equation:

$$y_k = H_k x_k + C_k \dot{x}_k + v_k \tag{34b}$$

in which $y_k \in R^m$ is the measurement (vector) value at time t_k; $x_k := x(t_k)$ (the generalized position at time t_k); $\dot{x}_k := \dot{x}_k(t_k)$ (the generalized velocity vector at time t_k); and $v_k \in R^m$ is a zero-mean, white, Gaussian noise sequence with covariance R_k, which is uncorrelated with the process noise $w(t)$. The same assumptions regarding the initial conditions are made here as in Sec. I [Eq. (1c)].

Compared with the continuous-time filter, which was presented in the preceding section, the continuous/discrete filter is composed of two separate computational stages,[29] namely, the time-update (propagation) stage and the measurement-up-

date stage. The time-update stage consists of propagating the estimation variables (estimated state and error covariance matrix) between two consecutive measurement epochs, whereas in the measurement-update stage these variables are updated (instantly) using the new information acquired in the most recent measurement. To describe the resulting algorithm, we break the following development into two parts, corresponding to the two stages of the filter. We start with the time-update stage, which, as will be shown in the sequel, can be derived directly using the results of the preceding section.

A. Time-Update Stage

Using the definitions of the matrices E, F, and $\bar{G}$, the propagation equation for the estimation error covariance matrix is[27,28]:

$$\dot{EPE}^T = FPE^T + EPF^T + \bar{G}Q\bar{G}^T \tag{35}$$

The problem of the V-lambda time update is stated below.

Let $P = V\Lambda V^T$ be the spectral decomposition of the estimation error covariance, where V is the matrix whose columns are eigenvectors of P, and Λ is the diagonal matrix of eigenvalues. Let $V_{k/k}$ and $\Lambda_{k/k}$ be the a posteriori spectral factors of $P_{k/k}$, the covariance at time t_k given measurements up to and including time t_k. Given $V_{k/k}$ and $\Lambda_{k/k}$, the problem is to directly compute the a priori spectral factors $V_{k+1/k}$ and $\Lambda_{k+1/k}$ at time t_{k+1} (given measurements up to and including time t_k) without resorting to an explicit computation of the covariance.

Recognizing the fact that the solution of the (linear) Lyapunov-type equation (35) is a special case of the solution of the (nonlinear) Riccati-type equation (6), the results of the preceding section can be readily applied to the current problem. The resulting time update algorithm, which is an adaptation of the continuous SR filter, will be presented.

The a priori $Z_{k+1/k}$ and $\Lambda_{k+1/k}$ factors (where Z is the weighted eigenvector matrix) at time t_{k+1} are obtained by integration of the following differential equations:

$$\dot{Z} = Z\Omega \tag{36a}$$

$$\dot{\Lambda} = \Gamma \tag{36b}$$

from time t_k, with the following initial conditions

$$Z(t_k) = Z_{k/k} \tag{37a}$$

$$\Lambda(t_k) = \Lambda_{k/k} \tag{37b}$$

where the a posteriori weighted eigenvector matrix $Z_{k/k}$ is computed by

$$Z_{k/k} = E^{-T}V_{k/k} \tag{38}$$

In Eq. (36) the matrices Γ and Ω are defined in a way similar to the definitions (24) and (25), with the following obvious modification of the matrix-valued function T_{iq} (used to calculate the scalar functions γ_{iq}):

$$T_{iq} := \begin{pmatrix} 0 & (\lambda_i M - \lambda_q K)^T \\ \lambda_q M - \lambda_i K & -(\lambda_i DM^T + \lambda_q MD^T) + GQG^T \end{pmatrix} \tag{39}$$

To obtain the time-update algorithm for the state estimate, the standard continuous-time KF algorithm is applied to the system equation (34) to get

$$M\ddot{\hat{x}} + D\dot{\hat{x}} + K\hat{x} = \mathbf{0} \tag{40}$$

which is integrated from t_k to t_{k+1}. The initial condition $\hat{x}(t_k)$ is the a posteriori state estimate at t_k:

$$\hat{x}(t_k) = \hat{x}_{k/k} \tag{41}$$

At time t_{k+1} the integration results in $\hat{x}(t_{k+1})$, which is the required a priori state estimate:

$$\hat{x}_{k+1/k} = \hat{x}(t_{k+1}) \tag{42}$$

Next we augment the time-update algorithm with a discrete-time measurement-update algorithm, thus completing the new SR estimator.

B. Measurement-Update Stage

Defining the augmented measurement matrix $\bar{H}_k$ as

$$\bar{H}_k := [H_k \,|\, C_k] \tag{43}$$

the measurement equation (34b) can be rewritten as

$$y_k = \bar{H}_k q_k + v_k \tag{44}$$

where the augmented state q_k is defined in Eq. (4). The measurement-update problem is as follows.

Given the a priori factors $\Lambda_{k+1/k}$ and $Z_{k+1/k}$ (resulting from the time-update integration described in part A of this section), where $\Lambda_{k+1/k}$ is the diagonal matrix of eigenvalues at time t_{k+1} given measurements up to and including t_k and $Z_{k+1/k}$ is the corresponding weighted eigenvector matrix, compute the a posteriori SR factors of the covariance (namely: $\Lambda_{k+1/k+1}$ and $V_{k+1/k+1}$) without computing explicitly the covariance $P_{k+1/k}$ (so that the SR numerical characteristics of the method will not be lost).

The algorithmic solution to the measurement-update problem is summarized in the theorem stated and proved below.

Theorem 2. Measurement update of the spectral factors. Given the time-propagated factors $\Lambda_{k+1/k}$ and $Z_{k+1/k}$, the augmented measurement matrix $\bar{H}_{k+1}$, and the nonsingular measurement noise covariance R_{k+1}, define the augmented matrix B_{k+1} as follows:

$$B_{k+1} := [E^T Z_{k+1/k}\Lambda_{k+1/k}^{-1/2} \,|\, \bar{H}_{k+1}^T R_{k+1}^{-1/2}] \tag{45}$$

and perform a singular value decomposition (SVD) of it:

$$B_{k+1} = Y_{k+1}[\Sigma_{k+1} \,|\, 0]U_{k+1}^T \tag{46}$$

Then the measurement updated spectral factors $V_{k+1/k+1}$ and $\Lambda_{k+1/k+1}$ are obtained readily as

$$V_{k+1/k+1} = Y_{k+1} \tag{47a}$$

$$\Lambda_{k+1/k+1}^{1/2} = \Sigma_{k+1}^{-1} \tag{47b}$$

In the SVD [Eq. (46)], Y_{k+1} is the orthogonal matrix whose columns are the eigenvectors of $B_{k+1}B_{k+1}^T$, Σ_{k+1} is the diagonal singular value matrix, and U_{k+1} is the orthogonal matrix whose columns are the eigenvectors of $B_{k+1}^T B_{k+1}$.

Proof. Write the measurement update equation in the "information form":

$$P_{k+1/k+1}^{-1} = P_{k+1/k}^{-1} + \bar{H}_{k+1}^T R_{k+1}^{-1}\bar{H}_{k+1} \tag{48}$$

Using the spectral factors of the covariance and the relation between the eigenvector matrix V and the weighted eigenvector matrix Z [Eq. (14)], Eq. (38) can be written as

$$V_{k+1/k+1}\Lambda_{k+1/k+1}^{-1}V_{k+1/k+1}^T = E^T Z_{k+1/k}\Lambda_{k+1/k}^{-1}Z_{k+1/k}^T E + \bar{H}_{k+1}^T R_{k+1}^{-1}\bar{H}_{k+1} \tag{49}$$

Using the definition of B_{k+1} from Eq. (45) in Eq. (49) yields

$$V_{k+1/k+1}\Lambda_{k+1/k+1}^{-1}V_{k+1/k+1}^T = B_{k+1}B_{k+1}^T \tag{50}$$

Now replace B_{k+1} in Eq. (50) by its SVD factors [Eq. (46)] to obtain

$$V_{k+1/k+1}\Lambda^{-1}_{k+1/k+1}V^T_{k+1/k+1} = Y_{k+1}\Sigma^2_{k+1}Y^T_{k+1} \tag{51}$$

from which the theorem follows.

Note that the update algorithm is based only on the SVD technique, for which extremely reliable and efficient algorithms exist today.[30,31]

To complete the measurement update algorithm, we need to specify how the state estimate is updated. This is performed via the ordinary KF algorithm, i.e.,

$$\hat{q}_{k+1/k+1} = \hat{q}_{k+1/k} + K_{k+1}(y_{k+1} - \bar{H}_{k+1}\hat{q}_{k+1/k}) \tag{52}$$

where K_{k+1} is the Kalman gain, which is computed (using the SR factors) according to the following algorithm:

$$K_{k+1} = S_{k+1/k+1}M_{k+1/k+1}R^{-1}_{k+1} \tag{53}$$

where $S_{k+1/k+1}$ and $M_{k+1/k+1}$ are defined as

$$S_{k+1/k+1} = V_{k+1/k+1}\Lambda^{1/2}_{k+1/k+1} \tag{54}$$

$$M_{k+1/k+1} = S^T_{k+1/k+1}\bar{H}^T_{k+1} \tag{55}$$

Having obtained the measurement update algorithm, a complete state estimator is formed by combining the time-update algorithm of part A with the measurement-update algorithm of this part. Before doing that, however, we note that up to this point the special orthogonality property of the eigenvector matrix was not exploited in our algorithm in any way. By using this property it may be possible in some cases to enhance the numerical characteristics of the estimator, and this is the subject of the next section.

IV. Orthogonalization of the Eigenvector Matrix

The time-update algorithm, described in the preceding section, consists of the numerical integration of nonlinear differential equations for the eigenvalue matrix Λ and for the weighted eigenvector matrix Z [Eqs. (36)]. Integrating Eqs. (36) with the initial conditions [Eqs. (37)] provides the a priori factors $\Lambda_{k+1/k}$ and $Z_{k+1/k}$. In certain cases, where it is prohibitive to use high-precision software for performing the task of the numerical integration, the need may arise to incorporate some measures to bound the numerical integration error introduced during the time-update process (note, in this regard, that the measurement-update algorithm may be considered "error-free" relative to the time-update algorithm because of its reliance on the SVD). Using similar arguments, it may be concluded that the continuous-time filtering algorithm, introduced in Sec. II, also may be sensitive to integration error (when low-accuracy integration methods are used). Thus, there too, a relatively simple means of bounding the integration error may be of great value.

In this section we show how the orthogonality of the eigenvectors of the covariance matrix may be used to reduce the integration error, thus exploiting the choice of the spectral factors to be the SR factors of the filter. Although, in the following discussion we will specifically address the time-update algorithm, it should be clear that the results of the present derivation apply also to the continuous-time filter, from which the time-update algorithm was derived.

The problem at hand can be described mathematically as follows: Suppose that at time t_{k+1}, after the numerical integration involved in the time update stage is performed, the a priori weighted eigenvector matrix obtained is $\tilde{Z}_{k+1/k}$. Now, because of the orthogonality of the covariance eigenvectors and the relation (14), the "true" matrix (i.e., the matrix that would have been computed using an infinite word length) $Z_{k+1/k}$ should satisfy

$$Z^T_{k+1/k}EE^TZ_{k+1/k} = I \tag{56}$$

where I is the identity matrix. Thus, $Z_{k+1/k}$ is said to be orthogonal with respect to the matrix EE^T. However, because of computation errors

$$\tilde{Z}^T_{k+1/k}EE^T\tilde{Z}_{k+1/k} \neq I$$

The problem is to find a matrix $\hat{Z}_{k+1/k}$ that satisfies the weighted orthogonality condition (56) and is "closest" to $\tilde{Z}_{k+1/k}$ in the sense that it minimizes the weighted least-squares criterion

$$J = ||E^T(\hat{Z}_{k+1/k} - Z)|| \tag{57}$$

where $||\bullet||$ denotes (here and in the sequel) the Frobenius (or Euclidean) matrix norm.

This problem has been addressed by several researchers in a variety of applications. Of all currently available solutions, we have chosen to use two algorithmically different methods, although, obviously, other algorithms may be more suitable for the particular implementation.

A. Orthogonalization Using the Singular Value Decomposition

Employing Eq. (14), Eq. (57) can be rewritten in the form

$$J = ||E^T\hat{Z}_{k+1/k} - V||$$

which has to be minimized subject to

$$V^TV = I$$

In this form, the problem is recognized to be a special case of the orthogonal Procrustes problem, which is stated below for $A \in R^{m,p}$, $B \in R^{m,p}$, and $Q \in R^{p,p}$:

$$\text{minimize } ||A - BQ||$$

$$\text{subject to } Q^TQ = I$$

The solution of the orthogonal Procrustes problem,[26] which uses the SVD, therefore can be readily applied to our orthogonalization problem, yielding the following algorithm.

Algorithm 3. Weighted orthogonalization using the SVD. Using the time-updated weighted eigenvector matrix $\tilde{Z}_{k+1/k}$, compute the eigenvector matrix

$$\tilde{V}_{k+1/k} = E^T\tilde{Z}_{k+1/k} \tag{58}$$

and perform a singular value decomposition of this product to obtain

$$\tilde{V}_{k+1/k} = X_{k+1}\Xi_{k+1}Y^T_{k+1} \tag{59}$$

Then the optimal eigenvector matrix is computed according to

$$\hat{V}_{k+1/k} = X_{k+1}Y^T_{k+1} \tag{60}$$

and, if needed, the optimal weighted eigenvector matrix is

$$\hat{Z}_{k+1/k} = E^{-T}\hat{V}_{k+1/k} \tag{61}$$

Remark. The orthogonalization algorithm 3 should be considered as a corrective step that is to be used optionally, immediately after the time-update stage (and before the measurement update). Note, however, that in the measurement-update stage the matrix $V_{k+1/k}$ should be used rather than $Z_{k+1/k}$ [see the definition of the augmented matrix B_{k+1}; Eq. (45)]. Hence, the orthogonalized $\hat{V}_{k+1/k}$ (rather than $\hat{Z}_{k+1/k}$) is used in the measurement-update stage that immediately follows the optional orthogonalization step, which means that the mass matrix inversion in Eq. (61) is not performed in most practical situations.

Since the orthogonalization method just presented is based on the SVD as the computational tool, it is reasonable to expect excellent results when the method is used. However, because the SVD may be computationally expensive, another method is presented, which, though expected to be inferior to the SVD regarding accuracy, demands less computational resources from the computer (and thus may be used more frequently to compensate for the lower accuracy).

B. Orthogonalization Using an Iterative Approach

The orthogonalization problem of this section was also addressed, in a different context, by Baruch and Bar-Itzhack.[25] In their work, the problem was to find a weighted orthogonal modal matrix that is closest, in the least-squares sense of Eq. (57), to a measured modal matrix (which, because of measurement errors, does not satisfy the orthogonality condition). Adopting the solution presented in Ref. 25, we obtain that in our case the optimal orthogonal matrix is

$$\hat{Z}_{k+1/k} = \tilde{Z}_{k+1/k}(\tilde{Z}^T_{k+1/k} EE^T \tilde{Z}_{k+1/k})^{-1/2} \tag{62}$$

However, in Eq. (62) we have to compute an inverse of a Square Root of a matrix, which may present a numerical problem by itself. To circumvent this problem, it was shown in Ref. 25 that an alternative way to compute the optimal $\hat{Z}_{k+1/k}$ is to perform it iteratively using the following recursion.

Algorithm 4. Iterative weighted orthogonalization.

$$\hat{Z}^{(i+1)}_{k+1/k} = \tfrac{1}{2}\hat{Z}^{(i)}_{k+1/k}[3I - \hat{Z}^{(i)T}_{k+1/k} EE^T \hat{Z}^{(i)}_{k+1/k}]$$

$$\hat{Z}^{(0)}_{k+1/k} = \tilde{Z}_{k+1/k} \tag{63}$$

where $\hat{Z}^{(i)}_{k+1/k}$ is the value of $\hat{Z}_{k+1/k}$ at the ith iteration, and the process is stopped when a predetermined desired level of orthogonality, μ, is achieved:

$$||\hat{Z}^{(i)T}_{k+1/k} EE^T \hat{Z}^{(i)}_{k+1/k} - I|| \leq \mu \tag{64}$$

We note here that usually $\tilde{Z}_{k+1/k}$ is very close to $Z_{k+1/k}$, the "true" matrix. This observation can be justified by recognizing that the only source of discrepancy between $\tilde{Z}_{k+1/k}$ and $Z_{k+1/k}$ is, by definition, the integration error corresponding solely to the most recent time update (all previous time updates being taken care of by preceding orthogonalization steps). Now, in our case, since $\tilde{Z}_{k+1/k}$ is assumed to be close enough to $Z_{k+1/k}$, it is reasonable to anticipate that, in many cases, one step of the recursive process [Eq. (63)] will result in a matrix $\hat{Z}_{k+1/k}$, which will satisfy the orthogonality condition [Eq. (64)].

Noting again that, in the measurement-update stage, which immediately follows the corrective orthogonalization, $\hat{V}_{k+1/k}$ is used (rather than $\tilde{Z}_{k+1/k}$), we may rewrite Eq. (63) in terms of the eigenvector matrix:

$$\hat{V}^{(i+1)}_{k+1/k} = \tfrac{1}{2}\hat{V}^{(i)}_{k+1/k}[3I - \hat{V}^{(i)T}_{k+1/k}\hat{V}^{(i)}_{k+1/k}] \tag{65a}$$

$$\hat{V}^{(0)}_{k+1/k} = \tilde{V}_{k+1/k} \tag{65b}$$

(Note that, if $\tilde{V}_{k+1/k}$ is orthogonal, then Eq. (65a) yields $\hat{V}_{k+1/k} = \tilde{V}_{k+1/k}$, as it should.) In practice, this corrective step was shown to yield excellent results in all numerical tests performed (though a rigorous analysis of its effect on the whole estimation scheme is still needed).

The complete continuous/discrete estimation algorithm, including the two alternatives for the weighted orthogonalization step, is summarized for the reader's convenience in Table 1.

V. Numerical Example

In this section we present the results of a numerical simulation of the continuous-time second-order filter, both in the conventional Kalman formulation and in the V-Lambda SR formulation. This simulation serves to demonstrate the superior numerical characteristics of the SR algorithm in a typical estimation problem. The system examined has the following parameter matrices:

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad D = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad K = \begin{pmatrix} 100 & 50 \\ 50 & 200 \end{pmatrix}$$

$$G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The measurement matrices are

$$H_1 = [1 \quad 0] \qquad H_2 = [1 \quad 0]$$

and the noise intensities are

$$Q = \text{diag}\{10^{-6}, 10^{-6}\}, \qquad R = 60$$

The initial estimation error covariance matrix for the augmented state vector was chosen as

$$P_0 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ & 2 & 0 & 1 \\ & & 3 & 0 \\ \text{SYM} & & & 4 \end{bmatrix}$$

In the KF algorithm three coupled Riccati-type matrix differential equations must be solved for the estimation error covariance matrix.[11] Defining the partition

$$P = \begin{pmatrix} \Sigma & S \\ S^T & U \end{pmatrix} \tag{66}$$

these equations are

$$\dot{\Sigma} = S + S^T - (H_1\Sigma + H_2 S^T)^T R^{-1}(H_1\Sigma + H_2 S^T) \tag{67a}$$

$$\dot{S}M^T = UM^T - (K\Sigma + DS^T)^T - (H_1\Sigma + H_2 S^T)^T R^{-1}(H_1 S + H_2 U)M^T \tag{67b}$$

$$M\dot{U}M^T = -(KS + DU)M^T - M(KS + DU)^T - M(H_1 S + H_2 U)^T R^{-1}(H_1 S + H_2 U)M^T + GQG^T \tag{67c}$$

where the initial condition matrices Σ_0, S_0, and U_0 are obtained from a partition of P_0 corresponding to Eq. (66). The solution obtained from these equations was compared to the SR solution obtained from Eqs. (26), using two different ordinary differential equation solvers:

1) IVPAG routine of IMSL Math/Library Version 1.0 (this routine uses an implicit Adams-Moulton method of order 12).

2) A fixed-step fourth-order Runge-Kutta routine (user-supplied). This routine was used to investigate the effect of the different orthogonalization schemes on the V-Lambda estimation algorithm when implemented employing nonsophisticated software.

All runs were performed in VMS FORTRAN on a DEC VAX 8650 machine at the computing center of State University of New York at Buffalo. The results obtained will be described in the sequel according to the integration method used.

A. Adams-Moulton Integration (High Accuracy, ISML Routine)

Using IMSL routine IVPAG, the integration accuracy is controlled via an error parameter TOL, which is specified by the user. The routine attempts to bound the integration error such that the global error will be proportional to TOL. The

integration error is defined as

$$\epsilon = [\Sigma(\epsilon_i^2)]^{1/2}$$

where ϵ_i is the absolute value of an estimate of the error in Y_i, the ith integration variable. Noting this definition, it is clear that using a smaller error parameter (TOL) with this routine is equivalent to requiring a more accurate solution. Several runs were performed, in which the conventional KF algorithm [Eq. (67)] was run in single-precision (SP) and in double-precision (DP), and the V-Lambda algorithm was run in SP. All simulations were carried from $t = 0$ to $t = 50$ s, when the results were compared. Although the error covariance matrix is not used explicitly in the V-Lambda algorithm, it was computed at $t = 50$ s to provide a basis for comparison with the conventional algorithm (note that this means that the covariance computed this way is *less* accurate than the eigenfactors from which it is computed). When the KF equations were solved using single-precision and TOL = 10^{-6}, the solution diverged shortly after the integration was started. The computed covariance started exhibiting negative diagonal entries (a theoretical impossibility) at $t = 10$ s, and the solution completely lost numerical significance thereafter. Trying to obtain higher accuracy using a smaller error bound (TOL) resulted in even worse behavior of the solution. Using the V-Lambda filter (in SP), no such problems were observed. Solution could be obtained for any error parameter, and for TOL = 10^{-7} the error covariance matrix [computed using Eq. (30)] at $t = 50$ s was

$$P_{V-\Lambda/SP} = \begin{bmatrix} 6.66877E-09 & -3.32410E-09 & -3.02009E-12 & -9.23360E-12 \\ & 6.66876E-09 & -8.68409E-12 & -2.71750E-12 \\ \text{SYM} & & 5.00584E-07 & 1.03040E-09 \\ & & & 5.00582E-07 \end{bmatrix} \tag{68}$$

This solution was next compared to a DP integration of the conventional KF algorithm. At TOL = 10^{-7}, the DP version of the KF algorithm behaved much like the SP version, yielding negative variances after $t = 13$ s. Only by decreasing the value of TOL to 10^{-11} could a satisfactory solution be obtained, yielding at $t = 50$ s the following covariance matrix:

$$P_{KF/DP} = \begin{bmatrix} 6.66650D-09 & -3.33350\text{E}-09 & 4.75571D-12 & 4.75578D-12 \\ & 6.66650\text{D}-09 & 4.75564D-12 & 4.75571D-12 \\ \text{SYM} & & 5.00021D-07 & 2.06378D-11 \\ & & & 5.00021D-07 \end{bmatrix} \tag{69}$$

Comparing the V-Lambda (SP, low-accuracy) solution [Eq. (68)] with the KF (DP, high-accuracy) solution [Eq. (69)] demonstrates, in this particular example, the numerical robustness and enhanced accuracy of the new estimation algorithm. Next, the effect of the orthogonalization algorithms detailed in Sec. IV was investigated. The results of this investigation are described in the sequel.

B. Fourth-Order Runge-Kutta Integration (Low-Accuracy, User-Supplied Routine)

To study the effect of the different orthogonalization schemes on the numerical integration process, a simple, fixed-step, Runge-Kutta routine was written to simulate the implementation of the algorithm in a computing environment that cannot support state-of-the-art mathematical software. Using this kind of routine, the integration accuracy is controlled via the step size, δt, which is specified by the user at the beginning of the process. Again, the KF equations [Eq. (67)] were solved using SP and DP arithmetic, whereas the V-Lambda equations were integrated using SP wordlength, with and without corrective orthogonalization.

When the conventional KF algorithm was used in SP with $\delta t = 2.5\,E-5$, the results obtained were very similar to those obtained using the KF/SP method with the IVPAG routine of IMSL: negative variances appeared at $t = 17$ s, and the solution lost numerical significance thereafter. Using the same KF algorithm with DP, however, produced excellent results even with a much larger step size of $\delta t = 10^{-3}$. At $t = 50$ s, the following error covariance was thus obtained:

$$P_{KF/DP} = \begin{bmatrix} 6.66667D-09 & -3.33333D-09 & 3.71658D-19 & 1.16567D-17 \\ & 6.66667D-09 & -1.20254D-17 & 9.34849D-20 \\ \text{SYM} & & 5.00000D-07 & -2.13036D-16 \\ & & & 5.00000D-07 \end{bmatrix} \tag{70}$$

When the V-Lambda algorithm was computed with the same step size used for the KF/SP solution, the following covariance matrix was computed at $t = 50$:

$$P_{V-\Lambda/SP} = \begin{bmatrix} 6.66159E-09 & -3.31950E-09 & 8.31620E-12 & -1.01053E-11 \\ & 6.66190E-09 & -1.82303E-11 & 4.38421E-12 \\ \text{SYM} & & 5.00304E-07 & 1.12627E-09 \\ & & & 5.00295E-07 \end{bmatrix} \tag{71}$$

Table 1 Continuous/discrete state estimator for large space structures

System model	$M\ddot{x} + D\dot{x} + Kx = Gw$	$E\{w\} = \mathbf{0}$	$E\{w(t)w(s)^{\mathrm{T}}\} = Q(t)\delta(t-s)$
Measurement model	$y_k = H_k x_k + C_k \dot{x}_k + v_k$	$E\{v_k\} = \mathbf{0}$	$E\{v_j v_k^T\} = R_k \delta_{jk}$
Initial conditions	$E\left\{\begin{pmatrix} x(0) \\ \dot{x}(0) \end{pmatrix}\right\} = \begin{pmatrix} x_0 \\ \dot{x}_0 \end{pmatrix}$		$E\left\{\begin{pmatrix} x_0 \\ \dot{x}_0 \end{pmatrix}^{(x_0^T \dot{x}_0^T)}\right\} = P_0$
State estimate time update	$M\ddot{\hat{x}}(t) + D\dot{\hat{x}}(t) + K\hat{x}(t) = \mathbf{0}$	$t \in [t_k, t_{k+1}]$	
	Integration initial conditions: $\begin{pmatrix} \hat{x} \\ \dot{\hat{x}} \end{pmatrix}_{@t_k} = \begin{pmatrix} \hat{x}_{k/k} \\ \dot{\hat{x}}_{k/k} \end{pmatrix}$		
	Integration results at t_{k+1}: $\begin{pmatrix} \hat{x}_{k+1/k} \\ \dot{\hat{x}}_{k+1/k} \end{pmatrix} = \begin{pmatrix} \hat{x} \\ \dot{\hat{x}} \end{pmatrix}_{@\, t_{k+1}}$		
Λ-Z time update	$\dot{\Lambda}(t) = \Gamma(t), \qquad \dot{Z}(t) = Z(t)\Omega(t); \qquad t \in [t_k, t_{k+1}]$		
	$\Omega_{qi} = \begin{cases} \dfrac{\gamma_{iq}}{\lambda_i - \lambda_q} & \text{for } \left\lvert \dfrac{\gamma_{iq}}{\lambda_i - \lambda_q} \right\rvert < \Omega_{\max} \\ 0 & \text{for } \lambda_i = \lambda_q \\ \Omega_{\max} \operatorname{sign}\left\{ \dfrac{\gamma_{iq}}{\lambda_i - \lambda_q} \right\} & \text{for } \left\lvert \dfrac{\gamma_{iq}}{\lambda_i - \lambda_q} \right\rvert \geq \Omega_{\max} \end{cases}$		
	$\Gamma := \operatorname{diag}[\gamma_{11}, \gamma_{22}, \ldots, \gamma_{nn}]$		
	$\gamma_{iq} := z_q^T T_{iq} z_i \qquad Z = [z_1 \vert z_2 \vert \ldots \vert z_n]$		
	$T_{iq} := \begin{pmatrix} 0 & (\lambda_i M - \lambda_q K)^T \\ \lambda_q M - \lambda_i K & -(\lambda_i D M^T + \lambda_q M D^T) + GQG^T \end{pmatrix}$		
	Initial conditions: $\Lambda(t_k) = \Lambda_{k/k}; \qquad Z(t_k) = E^{-T} V_{k/k}$		
	where $E = \operatorname{diag}\{I, M\}$		
	Integration results: $\Lambda_{k+1/k} = \Lambda(t_{k+1}); \qquad \bar{Z}_{k+1/k} = Z(t_{k+1})$		
Orthogonalization	**A. SVD**		**B. Iterative**
	$\tilde{V}_{k+1/k} = E^T \bar{Z}_{k+1/k}$		$\hat{V}_{k+1/k}^{(0)} = E^T \bar{Z}_{k+1/k}$
	$\tilde{V}_{k+1/k} = X_{k+1} \Xi_{k+1} Y_{k+1}^T$		$\hat{V}_{k+1/k}^{(j+1)} = \tfrac{1}{2} \hat{V}_{k+1/k}^{(j)} [3I - \hat{V}_{k+1/k}^{(j)T} \hat{V}_{k+1/k}^{(j)}]$
	$\hat{V}_{k+1/k} = X_{k+1} Y_{k+1}^T$		stop when
			$\lvert\lvert \hat{V}_{k+1/k}^{(j)T} \hat{V}_{k+1/k}^{(j)} - I \rvert\rvert \leq \mu$
Measurement update of the spectral factors	$B_{k+1} := [E^T Z_{k+1/k} \Lambda_{k+1}^{-1/2} \vert \bar{H}_{k+1}^T R_{k+1}^{-1/2}]$		
	$B_{k+1} \xrightarrow{\text{SVD}} Y_{k+1} [\Sigma_{k+1} \vert 0] U_{k+1}^T$		
	Read: $V_{k+1/k+1} = Y_{k+1}; \qquad \Lambda_{k+1/k+1}^{1/2} = \Sigma_{k+1}^{-1}$		
State estimate measurement update	$\hat{q}_{k+1/k+1} = \hat{q}_{k+1/k} + K_{k+1}(y_{k+1} - \bar{H}_{k+1} \hat{q}_{k+1/k})$		
	where: $\hat{q} := (\hat{x}^T, \dot{\hat{x}}^T)^T$		
	Kalman gain: $K_{k+1} = S_{k+1/k+1} M_{k+1/k+1} R_{k+1}^{-1}$		
	$S_{k+1/k+1} = V_{k+1/k+1} \Lambda_{k+1/k+1}; \qquad M_{k+1/k+1} = S_{k+1/k+1}^T \bar{H}_{k+1}^T$		

which compares favorably with the KF/DP solution [Eq. (70)]. Because of the stiffness inherent in the problem, no solution could be obtained using the V-Lambda algorithm with a larger step size. However, when a corrective orthogonalization step was used, a good solution was obtained easily with a step size of $\delta t = 10^{-3}$. Both orthogonalization methods were used with the same strategy: every second (simulation time) an orthogonality criterion was computed according to

$$\mu = ||Z(t)^T EE^T Z(t) - I||$$

Whenever μ was larger than a prespecified value of 10^{-5}, orthogonalization was carried out. Using the iterative orthogonalization, the following covariance was computed at $t = 50$ s:

$$P_{V-\Lambda/SP} = \begin{pmatrix} 6.66670E-09 & -3.33336E-09 & -4.33892E-14 & 8.66677E-14 \\ & 6.66670E-09 & 4.88366E-14 & -6.20096E-14 \\ \text{SYM} & & 5.00002E-07 & -5.96856E-13 \\ & & & 5.00002E-07 \end{pmatrix} \quad (72)$$

Almost identical results were obtained using the orthogonalization method based on the SVD. These results demonstrate, for the example presented here, how the orthogonalization can be used to enhance the numerical characteristics of the V-Lambda algorithm.

VI. Conclusions

Two new SR filtering algorithms, a continuous one and a hybrid continuous/discrete one, were presented for systems modeled by second-order models, such as LSS. The two algorithms retain the structure (symmetry, sparsity) of the original system parameters (mass, damping, and stiffness matrices), thus enabling the exploitation of this structure in the algorithm implementation.

Being based on the spectral decomposition of the covariance matrix, the new algorithms provide their user with an insight into the estimation process. This special choice of SR factors also makes it possible to use the orthogonality property of the eigenvectors to enhance the filter accuracy.

The numerical robustness of the continuous-time algorithm was demonstrated via a simple numerical example. The orthogonalization methods developed were shown to greatly enhance the V-Lambda algorithm by reducing the integration error, thus enabling the use of simple, non-state-of-the-art integration routines for the production of excellent results.

Further investigation is needed to determine the precise nature of how the orthogonalization process affects the numerical integration. Also, other implicit state-space formulations exist that can be used instead of the particular formulation used in Eq. (3) (e.g., the symmetric state-space form used in Ref. 32, or other implicit forms in Ref. 7). The unique structure of each formulation can conceivably be exploited to increase the efficiency of the V-Lambda algorithms. Another area for future research is the application of SR algorithms for the estimation of modal-space state variables, which may be important in the control of LSS. These subjects are currently under investigation.

Acknowledgments

The first two authors acknowledge the support of National Science Foundation Grant MEA8112826 and Air Force Office of Scientific Research Grant AFOSR 820242. The third author acknowledges the support of the National Science Foundation (and the Air Force Office of Scientific Research) under Grant ECS84-06152. The first author gratefully acknowledges the interest of Drs. John L. Junkins and Jer-Nan Juang, who offered helpful suggestions.

References

[1]Likins, P. W., "The Application of Multivariable Control Theory to Spacecraft Attitude," *Proceedings of the IFAC 4th International Symposium on Multivariable Technological Systems,* New Brunswick, Canada, 1977, pp. 11–20.

[2]Kosut, R. L., Salzwedel, H., and Emami-Naeini, A., "Robust Control of Flexible Spacecraft," *Journal of Guidance, Control, and Dynamics,* Vol. 6, March–April 1983, pp. 104–111.

[3]Juang, J.-N., Horta, L. G., and Robertshaw, H. H., "A Slewing Control Experiment for Flexible Structures," *Journal of Guidance, Control, and Dynamics,* Vol. 9, Sept.–Oct. 1986, pp. 599–607.

[4]Kwakernaak, H. and Sivan, R., *Linear Optimal Control Systems,* Wiley-Interscience, New York, 1972.

[5]Pissanetzky, S., *Sparse Matrix Technology,* Academic, New York, 1984.

[6]Bunch, J. R. and Rose, D. J. (ed.), *Sparse Matrix Computations,* Academic, New York, 1976.

[7]Laub, A. J. and Arnold, W. F., "Controllability and Observability Criteria for Multivariable Linear Second-Order Models," *IEEE Transactions on Automatic Control,* Vol. AC-29, Feb. 1984, pp. 163–165.

[8]Hughes, P. C. and Skelton, R. E., "Controllability and Observability of Linear Matrix Second-Order Systems," *Journal of Applied Mechanics,* Vol. 47, June 1980, pp. 415–420.

[9]Hughes, P. C. and Skelton, R. E., "Controllability and Observability for Flexible Spacecraft," *Journal of Guidance and Control,* Vol. 3, Sept.–Oct. 1980, pp. 452–459.

[10]Bender, D. J. and Laub, A. J., "Controllability and Observability at Infinity of Multivariable Linear Second-Order Models," *IEEE Transactions on Automatic Control,* Vol. AC-30, Dec. 1985, pp. 1234–1237.

[11]Hashemipour, H. R. and Laub, A. J., "Kalman Filtering for Second-Order Models," *Journal of Guidance, Control, and Dynamics,* Vol. 11, March–April 1988, pp. 181–186.

[12]Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of ASME, Journal of Basic Engineering,* Vol. 82D, March 1960, pp. 34–45.

[13]Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," *Transactions of ASME, Journal of Basic Engineering,* Vol. 83, 1961, pp. 95–108.

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

[15]Potter, J. E. and Stern, R. G., "Statistical Filtering of Space Navigation Measurements," *Proceedings of the AIAA Guidance and Control Conference,* AIAA, New York, 1963.

[16]Oshman, Y. and Bar-Itzhack, I. Y., "Eigenfactor Solution of the Matrix Riccati Equation—A Continuous Square Root Algorithm," *IEEE Transactions on Automatic Control,* Vol. AC-30, Oct. 1985, pp. 971–978.

[17]Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation,* Academic, New York, 1977.

[18]Lawson, C. L. and Hanson, R. S., *Solving Least Squares Problems,* Prentice-Hall, Englewood Cliffs, NJ, 1974.

[19]Bierman, G. J., "Measurement Updating Using the U-D Factorization," *Automatica,* Vol. 12, July 1976, pp. 375–382.

[20]Thornton, C. L. and Bierman, G. J., "Gram-Schmidt Algorithms for Covariance Propagation," *Proceedings of the IEEE Conference on Decision and Control,* Institute of Electrical and Electronics Engineers, New York, 1975, pp. 489–498.

[21]Oshman, Y. and Bar-Itzhack, I. Y., "Square Root Filtering Via Covariance and Information Eigenfactors," *Automatica,* Vol. 22, No. 5, Sept. 1986, pp. 599–604.

[22]Oshman, Y. and Bar-Itzhack, I. Y., "Discrete Time Gain-Free V-Lambda Filtering," *Proceedings of the IEEE Conference on Decision and Control,* Institute of Electrical and Electronics Engineers, New York, 1986, pp. 1609–1611.

[23]Oshman, Y. and Bar-Itzhack, I. Y., "Square Root V-Lambda Filtering Using Normalized State Estimate," *Proceedings of the SIAM Conference on Linear Algebra in Signals, Systems and Control,* Society of Industrial and Applied Mathematics, Philadelphia, 1988, pp. 446–458.

[24]Ham, F. M. and Grover Brown, R., "Observability, Eigenvalues and Kalman Filtering," *IEEE Transactions on Aerospace and Electronic Systems,* Vol. AES-19, No. 2, March 1983, pp. 269–273.

[25]Baruch, M. and Bar-Itzhack, I. Y., "Optimal Weighted Orthogonalization of Measured Modes," *AIAA Journal,* Vol. 16, April 1978, pp. 346–351.

[26]Golub, G. H. and Van Loan, C. F., *Matrix Computations,* Johns Hopkins Univ. Press, Baltimore, MD, 1983, pp. 425–426.

[27]Arnold, W. F. and Laub, A. J., "Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations," *Proceedings of the IEEE,* Vol. 72, No. 12, Institute of Electrical and Electronics Engineers, New York, 1984, pp. 1746–1754.

[28]Laub, A. J., "Schur Techniques in Invariant Imbedding Methods for Solving Two-Point Boundary Value Problems," *Proceedings of the 21st IEEE Conference on Decision and Control,* Institute of Electrical and Electronics Engineers, New York, 1982, pp. 56–61.

[29]Gelb, A. (ed.), *Applied Optimal Estimation,* MIT, Cambridge, MA, 1974.

[30]Golub, G. H. and Reinsch, C., "Singular Value Decomposition and Least Squares Solutions," *Numerische Mathematik,* Vol. 14, 1970, pp. 403–420.

[31]Brent, R. P. and Luk, F. T., "The Solution of Singular Value and Symmetric Eigenvalue Problems on Mulitprocessor Arrays," *SIAM Journal on Scientific Statistical Computing,* Vol. 6, 1985, pp. 69–84.

[32]Craig, R. R., Jr. and Zhenhua, N., "Component Mode Synthesis for Model Order Reduction of Non-Classically-Damped Systems," AIAA Paper 87-2386, Aug. 1987.