Proceedings of the 1996 IEEE International
Conference on Control Applications
Dearborn, MI • September 15-18, 1996

Fixed-Interval Smoothing Algorithm Based on Singular Value Decomposition[1]

Youmin Zhang X. Rong Li

University of New Orleans
Department of Electrical Engineering
New Orleans, LA 70148
ymzee@uno.edu *xrlee@uno.edu*

Abstract

In this paper, a new fixed-interval smoothing algorithm based on singular value decomposition (SVD) is presented. The main idea of the new algorithm is to combine a forward-pass SVD-based square-root Kalman filter, developed recently by the authors, with a Rauch-Tung-Striebel (R-T-S) backward-pass recursive smoother by using the SVD as a main computational tool. Similarly to the SVD-based square-root filter, the proposed smoother has good numerical stability and does not require covariance matrix inversion. It is formulated in a vector-matrix form, and thus is handy for implementation with parallel computers. A typical numerical example is used to demonstrate the performance of the new smoother.

1. Introduction

The problem of fixed-interval, discrete-time linear smoothing has been addressed in many books [1-4] and papers [5-16]. Most of the algorithms appeared in the literature are in a non-factorized form, which rely in one way or another on the conventional Kalman filter. One of the earliest algorithms in this class is the Rauch-Tung-Striebel (R-T-S) fixed-interval smoother [5], which initiated the development of several other linear smoothing algorithms [9]. It is well-known that in the Kalman filter, the update covariance matrix $P(t)$ is sensitive to the truncation errors and there is no guarantee that $P(t)$ will always be symmetric and positive definite, especially for implementation in mini- or micro-computers with a short word length [2]. In order to solve this problem, various square root filtering algorithms, including the square-root covariance filter (SRCF) and the square root information filter (SRIF) as well as the U-D factorization filter, have been developed, in which the covariance matrix $P(t)$ recursion is replaced by numerically stable square-root or U-D factorization recursions [2-4]. Although these recursions avoid the costly covariance matrix inversion and a numerically hazardous difference of positive semi-definite matrices, they require a certain inversion of backward substitution steps, and none of them is particularly well-suited for parallel implementation. This rules out their application to a class of systems with singular transition matrices. SVD is one of the most stable and accurate matrix decomposition methods in numerical linear algebra and is easy to be implemented in parallel computers [17]. Because of these advantages, the SVD-based Kalman filter [24-28], recursive identification [29] have been developed in recent years. However, only a few papers deal with the fixed-interval smoothing problem.

This paper proposes a new factorized fixed-interval smoother based on the SVD-based Kalman filter developed recently by the authors [28]. The development of the new smoother is based on the R-T-S smoother formulation and makes use of the singular value decomposition of the covariance matrix into the $P = U\Lambda V^T$ form, where U and V are eigenvector matrices and Λ is a diagonal eigenvalue matrix. The new algorithm is numerically robust, dose not need the inversion of the covariance matrix, and is applicable to systems with singular transition matrices. Another advantage of the proposed smothering algorithm is that it is suitable for parallel implementation.

2. Singular Value Decomposition

One of the basic and most important tools of modern numerical analysis is the singular value decomposition (SVD). It has become a fundamental tool in linear algebra, system theory, and modern signal processing and control [17, 23]. It is because that not only the SVD permits an elegant problem formulation, but also it provides geometrical and algebraic insight together with numerically robust implementation. It includes the important eigenvalue decomposition of a Hermitian matrix as a special case. A well-known superiority of the SVD-based methods is that singular values can be computed more efficiently with much greater numerical stability than eigenvalues. Advantage of certain structural features of the SVD can also be taken of for filtering and smoothing algorithms as proposed in this paper, which can largely reduce computational requirement. For a survey of the theory and its many interesting applications, see, e.g., [17, 19, 23].

[1] Research partially supported by NSF (grant ECS-9496319), NNSFC (grant 69274015) and ASFC (grant 94E53186).

The singular value decomposition of an $m \times n$ matrix A $(m \geq n)$, is a factorization of A into product of three matrices. That is, there exist orthogonal matrices $U \in R^{m \times m}$ and $V \in R^{n \times n}$ such that

$$A = U \Lambda V^T, \qquad \Lambda = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \tag{1}$$

where $\Lambda \in R^{m \times n}$ and $S = diag(\sigma_1, ..., \sigma_r)$ with

$$\sigma_1 \geq \cdots \geq \sigma_r > 0.$$

The numbers $\sigma_1, ..., \sigma_r$ are called the singular values of A and are the positive square roots of the eigenvalues of $A^T A$. The columns of $U = [\mathbf{u}_1, ..., \mathbf{u}_m]$ are called the left singular vectors of A (the orthonormal eigenvectors of AA^T) and the columns of $V = [\mathbf{v}_1, ..., \mathbf{v}_n]$ the right singular vectors of A (the orthonormal eigenvectors of $A^T A$). The left and right singular vectors form a basis for the row-space and the column-space of A. If $A^T A$ is positive definite, then (1) reduces to

$$A = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^T \tag{2}$$

where S is an $n \times n$ diagonal matrix. Specifically, if A itself is symmetric and positive definite, we then have a symmetric singular value decomposition

$$A = USU^T = UD^2U^T \tag{3}$$

The standard method for computing (1) is the Golub-Kahan-Reisch (G-K-R) SVD algorithm [17], in which a Householder transformation is used to bidiagonalize the given matrix and the singular values of the resultant bidiagonal form are computed using QR method. Recently, with the advent of massively parallel computer architecture, two classical SVD methods, Hestenes algorithm (one-sided Jocabi) [20] and Kogbetliantz algorithm (two sided-Jacabi) [21], have gained a renewed interest for their inherent parallelism and vectorizability.

By comparing the computing flops of the G-K-R algorithm with Hestenes algorithm and Kogbetliantz algorithm, it can be seen that the G-K-R algorithm is most computationally efficient on a sequential machine. It becomes less attractive, however, on a parallel processor, where Hestenes algorithm and Kogbetliantz algorithm are important.

Our present smoother formulation is based on G-K-R algorithm and runs on a sequential IBM compatible personal computer.

3. R-T-S Smoother

The fixed-interval smoothing problem is to find $\{\hat{\mathbf{x}}_{k|N}\}_{k=0}^{N}$ given output data $\{\mathbf{y}_j\}_{j=0}^{N}$. In 1965, Rauch-Tung-Striebel [5] proposed a discrete-time, optimal fixed-interval smoothing algorithm, which is referred to as the R-T-S smoother later. The R-T-S smoother formulation is often cited and has been applied and improved by researchers because of its simplicity [6, 7, 9]. Its principal disadvantage lies in the required inversion of predicted covariance matrix $P_{k+1|k}$, as shown in (12) later, at each step. Not only is this computationally inefficient, but also it is numerically unstable generally [7, 9].

Consider the following discrete-time linear system described by:

$$\begin{cases} \mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{w}_k \\ \mathbf{y}_k = H_k \mathbf{x}_k + \mathbf{v}_k \end{cases} \tag{4}$$

where $k = 0, 1, ..., N$, $\mathbf{x}_k \in R^n$ is the state vector, $\mathbf{y}_k \in R^m$ is the measurement vector, $\mathbf{w}_k \in R^n$ and $\mathbf{v}_k \in R^m$ are process and measurement noises, respectively. The sequences $\{\mathbf{w}_k\}$ and $\{\mathbf{v}_k\}$ are assumed to be zero mean Gaussian white noise sequences as follows

$$E[\mathbf{w}_k] = 0, \; E[\mathbf{v}_k] = 0, \; E[\mathbf{w}_k \mathbf{w}_j^T] = Q_k \delta_{kj}, \; E[\mathbf{v}_k \mathbf{v}_j^T] = R_k \delta_{kj}$$

The initial state $\mathbf{x}_0$ is assumed to be a Gaussian random variable with mean μ_0 and covariance P_0. It is assumed that the process and measurement noise sequences and the initial state random vector are mutually uncorrelated.

Given the model (4), the Kalman filter formulation in covariance/information mode, used in the forward pass of the smoother, is then described by

Prediction:

$$\hat{\mathbf{x}}_{k+1|k} = \Phi_k \hat{\mathbf{x}}_{k|k} \tag{5}$$

$$P_{k+1|k} = \Phi_k P_{k|k} \Phi_k^T + Q_k \tag{6}$$

Update:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1}(\mathbf{y}_{k+1} - H_{k+1}\hat{\mathbf{x}}_{k+1|k}) \tag{7}$$

$$P_{k+1|k+1}^{-1} = P_{k+1|k}^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \tag{8}$$

$$K_{k+1} = P_{k+1|k+1} H_{k+1}^T R_{k+1}^{-1} \tag{9}$$

Then the R-T-S smoothing algorithm is described by

$$\hat{\mathbf{x}}_{k|N} = \hat{\mathbf{x}}_{k|k} + G_k(\hat{\mathbf{x}}_{k+1|N} - \hat{\mathbf{x}}_{k+1|k}) \tag{10}$$

$$P_{k|N} = P_{k|k} + G_k(P_{k+1|N} - P_{k+1|k})G_k^T \tag{11}$$

$$G_k = P_{k|k} \Phi_k^T P_{k+1|k}^{-1} \tag{12}$$

and the recursion is a backward sweep from $k = N$ down to $k = 0$.

Although the R-T-S smoother is often cited and applied, it has two shortcomings. One is that the smoother gain, given by (12), involves a computationally burdensome covariance matrix inversion. The other is that the smoother covariance recursion, (11), involves a difference of positive (semi-)definite matrices that is subject to numerical instability. In order to solve these problems, Bierman (1983) [9] proposed a computationally efficient sequential fixed-interval smoother by using the U-D factorization. This algorithm avoids both shortcomings of the R-T-S formulation and is at the same time computationally more efficient. More recently, Park and Kailath (1995 & 1996) [15, 16] developed a new square-root R-T-S smoother which uses combined square-root array form and avoid matrix inversion and backsubstitution steps in forming the state estimates. An alternative is to use the singular value decomposition for the computation of covariance to obtain an SVD-based smoother. We derive briefly the SVD-based Kalman filter [28] in Section 4 and the derivation of a new SVD-based fixed-interval smoother is given in Section 5.

4. SVD-Based Kalman Filter

In the prediction covariance (6) of the Kalman filter, assume that the SVD of covariances $P_{k|k}$ is available for all t_k and has been propagated and updated by the filter algorithm. Thus, we have

$$P_{k|k} = U_{k|k} D_{k|k}^2 U_{k|k}^T \tag{13}$$

Equation (6) can thus be written as

$$P_{k+1|k} = \Phi_k U_{k|k} D_{k|k}^2 U_{k|k}^T \Phi_k^T + Q_k \tag{14}$$

We want to find the factors $U_{k+1|k}$ and $D_{k+1|k}^2$ such that $P_{k+1|k} = U_{k+1|k} D_{k+1|k}^2 U_{k+1|k}^T$, where $U_{k+1|k}$ is orthogonal and $D_{k+1|k}$ diagonal. Provided that there is no danger of deterioration in numerical accuracy, one could, in a brute force fashion, compute $P_{k+1|k}$ and then apply the SVD of symmetric positive definite matrix given by (3). It has been shown, however, that this is not a good way in numerical exercise [18]. Instead, if we define the following matrix

$$\begin{bmatrix} D_{k|k} U_{k|k}^T \Phi_k^T \\ \sqrt{Q_k^T} \end{bmatrix} \tag{15}$$

and compute its SVD, then we get

$$\begin{bmatrix} D_{k|k} U_{k|k}^T \Phi_k^T \\ \sqrt{Q_k^T} \end{bmatrix} = U_k' \begin{bmatrix} D_k' \\ 0 \end{bmatrix} (V_k')^T \tag{16}$$

Pre-multiplying both sides by their transposes, we have

$$\begin{aligned} &\Phi_k U_{k|k} D_{k|k}^2 U_{k|k}^T \Phi_k^T + \sqrt{Q_k}(\sqrt{Q_k})^T \\ = \;\; & V_k' \begin{bmatrix} (D_k')^T & 0 \end{bmatrix} (U_k')^T U_k' \begin{bmatrix} D_k' \\ 0 \end{bmatrix} (V_k')^T \end{aligned} \tag{17}$$

That is

$$\Phi_k U_{k|k} D_{k|k}^2 U_{k|k}^T \Phi_k^T + Q_k = V_k' D_k'^2 (V_k')^T \tag{18}$$

It follows from (14) that V_k' and D_k' are just the sought-after $U_{k+1|k}$ and $D_{k+1|k}$:

$$\begin{cases} U_{k+1|k} = V_k' \\ D_{k+1|k} = D_k' \end{cases} \tag{19}$$

Similarly to the above derivation of prediction equation, the update equation can also be acquired by using SVD. Applying the SVD of symmetric positive define matrix to $P_{k+1|k+1}$ and $P_{k+1|k}$ respectively, we may get

$$\begin{aligned} & P_{k+1|k+1}^{-1} \\ &= (U_{k+1|k+1} D_{k+1|k+1}^2 U_{k+1|k+1}^T)^{-1} \\ &= (U_{k+1|k} D_{k+1|k}^2 U_{k+1|k}^T)^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \\ &= (U_{k+1|k}^T)^{-1} D_{k+1|k}^{-2} U_{k+1|k}^{-1} + (U_{k+1|k}^T)^{-1} U_{k+1|k}^T \\ &\quad \cdot H_{k+1}^T R_{k+1}^{-1} H_{k+1} U_{k+1|k} U_{k+1|k}^{-1} \\ &= U_{k+1|k}^{-T} [D_{k+1|k}^{-2} + U_{k+1|k}^T H_{k+1}^T R_{k+1}^{-1} H_{k+1} U_{k+1|k}] U_{k+1|k}^{-1} \end{aligned} \tag{20}$$

Let $R_{k+1}^{-1} = L_{k+1} L_{k+1}^T$ in (20) be the Cholesky decomposition of the inverse of the covariance matrix. By constructing the matrix

$$\begin{bmatrix} L_{k+1}^T H_{k+1} U_{k+1|k} \\ D_{k+1|k}^{-1} \end{bmatrix} \tag{21}$$

and computing its SVD, we have

$$\begin{bmatrix} L_{k+1}^T H_{k+1} U_{k+1|k} \\ D_{k+1|k}^{-1} \end{bmatrix} = \bar{U}_{k+1}' \begin{bmatrix} \bar{D}_{k+1}' \\ 0 \end{bmatrix} (\bar{V}_{k+1}')^T \tag{22}$$

and thus

$$P_{k+1|k+1}^{-1} = (U_{k+1|k}^T)^{-1} \bar{V}_{k+1}' \bar{D}_{k+1}'^2 \bar{V}_{k+1}'^T U_{k+1|k}^{-1} \tag{23}$$

We then get

$$\begin{cases} U_{k+1|k+1} = U_{k+1|k} \bar{V}_{k+1}' = V_k' \bar{V}_{k+1}' \\ D_{k+1|k+1} = (\bar{D}_{k+1}')^{-1} \end{cases} \tag{24}$$

The filter gain matrix can be obtained as follows:

$$\begin{aligned} K_{k+1} &= P_{k+1|k+1} H_{k+1}^T R_{k+1}^{-1} \\ &= U_{k+1|k+1} D_{k+1|k+1}^2 U_{k+1|k+1}^T H_{k+1}^T R_{k+1}^{-1} \end{aligned} \tag{25}$$

There is no need to obtain a formula for the SVD of K_{k+1}, which can be acquired straightforward.

The state update is given by (7).

The equations (5), (16) - (19) for prediction and (7) and (22) - (25) for update describe an SVD-based square-root Kalman filter, which is the basis of our SVD-based smoothing algorithm.

Remark 1: The proposed algorithm requires only Λ and the right singular matrix V to be computed. Since the left singular matrix U need not be explicitly formed, the computational load is significant lower.

Remark 2: The computation of filter gain K_{k+1} and state estimates $\hat{\mathbf{x}}_{k+1|k}$, $\hat{\mathbf{x}}_{k+1|k+1}$ are straightforward. The essential calculation for K_{k+1}, $\hat{\mathbf{x}}_{k+1|k}$ and $\hat{\mathbf{x}}_{k+1|k+1}$ is the matrix-matrix and matrix-vector multiplications. Notice that L_k^T is triangular and $H_k^T L_k = (L_k H_k^T)^T$ can be obtained from the previous computation. The computational requirement can be reduced further by exploiting this property.

5. New SVD-Based Smoothing Algorithm

In the covariance equation (11) of the R-T-S smoothing algorithm, assume that the SVD of covariance $P_{k+1|N}$ is available for all k and has been propagated and updated by the smoothing algorithm. Thus, we have

$$P_{k+1|N} = U_{k+1|N} D^2_{k+1|N} U^T_{k+1|N} \tag{26}$$

Our goal is to find the factors $U_{k|N}$ and $D^2_{k|N}$ from (11) such that

$$P_{k|N} = U_{k|N} D^2_{k|N} U^T_{k|N} \tag{27}$$

where $U_{k|N}$ is orthogonal and $D_{k|N}$ is diagonal. Similarly to the derivation of the SVD-based Kalman filter presented in the preceding section, the SVD computation of symmetric positive definite matrix $P_{k|N}$ can be acquired as follows.

Define

$$E_k = P_{k|k} - G_k P_{k+1|k} G_k^T \tag{28}$$

It can be rewritten as

$$E_k = (I - G_k \Phi_k) P_{k|k} \tag{29}$$

Equation (11) can be written as

$$P_{k|N} = E_k + G_k P_{k+1|N} G_k^T \tag{30}$$

In order to carry out the SVD for (30), it is necessary to compute the SVD of E_k. By applying the matrix inversion lemma

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1} \tag{31}$$

to (28), we get, after some manipulation,

$$E_k^{-1} = P^{-1}_{k|k} + \Phi_k^T Q_k^{-1} \Phi_k \tag{32}$$

Define the singular value decomposition of matrix E_k as

$$E_k = U_k D_k^2 U_k^T \tag{33}$$

and consider the following matrix

$$\begin{bmatrix} \sqrt{Q_k^{-1}}\Phi_k \\ D^{-1}_{k|k} U^T_{k|k} \end{bmatrix} \tag{34}$$

By computing its SVD, we may get

$$\begin{bmatrix} \sqrt{Q_k^{-1}}\Phi_k \\ D^{-1}_{k|k} U^T_{k|k} \end{bmatrix} = U_k'' \begin{bmatrix} D_k'' \\ 0 \end{bmatrix} (V_k'')^T \tag{35}$$

Pre-multiplying both sides by their transposes, we have

$$P^{-1}_{k|k} + \Phi_k^T Q_k^{-1} \Phi_k = V_k''(D_k'')^2(V_k'')^T \tag{36}$$

Comparing (32), (33) and (36), we find

$$\left\{ \begin{array}{l} U_k = V_k'' \\ D_k = (D_k'')^{-1} \end{array} \right. \tag{37}$$

Now consider the following matrix

$$\begin{bmatrix} D_k U_k^T \\ D_{k+1|N} U^T_{k+1|N} G_k^T \end{bmatrix} \tag{38}$$

and compute its SVD

$$\begin{bmatrix} D_k U_k^T \\ D_{k+1|N} U^T_{k+1|N} G_k^T \end{bmatrix} = \bar{U}_k'' \begin{bmatrix} \bar{D}_k'' \\ 0 \end{bmatrix} (\bar{V}_k'')^T \tag{39}$$

Pre-multiplying both sides by its transpose, we have

$$U_k D_k^2 U_k^T + G_k P_{k+1|N} G_k^T = \bar{V}_k''(\bar{D}_k'')^2(\bar{V}_k'')^T \tag{40}$$

Comparing with (30), we get

$$\left\{ \begin{array}{l} U_{k|N} = \bar{V}_k'' \\ D_{k|N} = \bar{D}_k'' \end{array} \right. \tag{41}$$

In this manner, a new covariance update of the smoother is obtained. The crucial component of the update involves the computation of SVD. The smoother gain matrix is directly computed as

$$G_k = U_{k|k} D^2_{k|k} U^T_{k|k} \Phi_k^T U_{k+1|k} D^{-2}_{k+1|k} U^T_{k+1|k} \tag{42}$$

The smoothed state can be computed by (10).

6. Numerical Example

The system considered is described by (4) with

$$\Phi_k = \begin{bmatrix} 0.98267788 & -0.00279856 & -0.00247742 \\ -0.03010306 & 0.97903810 & -0.00136841 \\ 0.00214054 & -0.00897400 & 0.89097150 \end{bmatrix}$$

$$H_k = \begin{bmatrix} -0.83977904 & 0.20800722 & 0.00368785 \\ -0.74444975 & -0.10884998 & -0.01013156 \\ -0.67032373 & -0.28034698 & -0.02782092 \end{bmatrix}$$

and the process noise $\{\mathbf{w}_k\}$ and measurement noise $\{\mathbf{v}_k\}$ have covariances $Q_k = 10^{-5} I_{3\times3}$ and $R_k = I_{3\times3}$, respectively. The initial state has covariance $P_0 = 10^4 I_{3\times3}$.

The SVD-based Kalman filter was used for the forward state estimation, while the new smoother was used in the backward pass. The results of state estimation and mean square error of estimation by SVD-based Kalman filter with comparison to the conventional Kalman filter are given in Figure 1. The superiority of the new SVD-based filter is evident. The estimation results and RMS from our SVD-based smoothing algorithm and SVD-based filtering algorithm are shown in Figure 2. As expected, the new SVD-based filter and smoother have better estimation accuracy due to numerical robustness and fast convergency than conventional filter and smoother. Furthermore, comparing to the R-T-S smoother, the new smoother does not need to compute the inversion of covariance matrix, and makes full use of the symmetry and positive definite property of covariance matrix in the SVD computation. These make the algorithm not only numerical robust but also computationally efficient.

7. Conclusions

A new fixed-interval smoothing algorithm has been presented, which is based on the numerically robust SVD technique associated with forward-pass SVD-based square-root Kalman filter. The simulation results show that the new SVD-based Kalman filter and fixed-interval smoother has not only the better estimation accuracy, but also faster convergence. The computational requirement has been reduced by using symmetry and positive definite property of covariance matrix, and some special features of matrix operation for SVD computation. Further, the new algorithm does not require inversion of covariance matrix. As a result, the SVD-based square-root fixed-interval smoother is highly accurate and numerically robust.

References

[1] Anderson, B.D.O. and Moore, J.B., *Optimal Filtering*, Englewood Cliffs, NJ: Prentice-Hall, 1979.

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

[3] Maybeck, P.S., *Stochastic Models, Estimation and Control*, Vol.1-Vol.2, New York: Academic Press, 1979, 1982.

[4] Bar-Shalom, Y. and Li, X.R., *Estimation and Tracking: Principles, Techniques, and Software*, Boston, MA: Artech House, 1993.

[5] Rauch, N.E., Tung, F. and Striebel, C.T., Maximum Likelihood Estimates of Linear Dynamic Systems, *AIAA J.* Vol. 3, No. 8, 1965, 1445-1450.

[6] Kaminski, P.G., Bryson, A.E., Discrete Square Root Smoothing, Proc. *AIAA Guidance Contr. Conf.*, 1972.

[7] Meditch, J.S., A Survey of Data Smoothing for Linear Nonlinear Dynamic Systems, *Automatica*, Vol.9, 1973, 151-162.

[8] Bierman, G.J., Sequential Square Root Filtering and Smoothing of Discrete Linear Systems, *Automatica*, Vol. 10, No. 1, 1974, 147-158.

[9] Bierman, G.J. A New Computationally Efficient Fixed-Interval, Discrete-Time Smoother, *Automatica*, Vol.19, No.5, 1983, 503-511.

[10] Watanabe, K., A New Forward-Pass Fixed-Interval Smoother Using the UD Information Matrix Factorization, *Automatica*, Vol. 22, 1986, 465-475.

[11] Watanabe, K. and Tzafestas, S.G., New Computationally Efficient Formula for Backward-pass Fixed-interval Smoother and its UD Factorization Algorithm, *IEE Proc. Pt.-D*, Vol. 136, No. 2, 1989, 73-78.

[12] McReynolds, S.R., Covariance Factorization Algorithms for Fixed-Interval Smoothing of Linear Discrete Dynamic Systems, *IEEE Trans. on Auto. Contr.*, Vol. 35, No. 10, 1990, 1181-1183.

[13] Zhang Y.M. , Zhang, H.C., Dai, G.Z., A New Bias Partitioned Square-Root Information Filter and Smoother for Aircraft State Estimation, *Proc. of 31st IEEE Conf. on Decision and Control*, Tucson, 1992, 741-742.

[14] Zhang, Y.M., Dai G.Z. and Zhang H.C., New Development of Kalman Filtering Method, *J. Control Theory and Application*, Vol.12, No.6, 1995.

[15] Park, P. and Kailath, T., Square-Root RTS Smoothing Algorithms, *Int. J. Control*, Vol. 62, No. 5, 1995, 1049-1060.

[16] Park, P. and Kailath, T., New Square-Root Smoothing Algorithms, *IEEE Trans. Automat. Contr.*, Vol. 41, No. 5, 1996, 727-732.

[17] Golub, G.H. and Van Loan, C.F., *Matrix Computations*, Second Edition, London: The Johns Hopkins Press Ltd., 1989.

[18] Golub, G.H. and Reinsch, C., Singular Value Decomposition and Least Squares Solutions, *Numer. Math.*, Vol.14, 1970, 403-420.

[19] Klema, V.C. and Laub, A.J., The Singular Value Decomposition: Its Computation and Some Applications. *IEEE Trans. Automat. Contr.*, Vol. 25, No. 2, 1980, 164-176.

[20] Hestenes, M.R., Inversion of Matrix by Biorthogonation and Related Results, *J. Soc. Indust. Appl. Math.*, Vol.6, 1958, 51-90.

[21] Kogbetnantz, E., Solution of Linear Equations by Diagonalization of Coefficient Matrices, *Quart. Appl. Math.*, Vol. 13, 1955, 123-132.

[22] Berry, M. and Sameh, A., An Overview of Parallel Algorithms for the Singular Value and Symmetric Eigenvalue Problems, *J. Comput. Appl. Math.*, Vol. 27, 1989, 191-213.

[23] Van der Veen. A, Deprettere, E.F. and Swindlehurs, A.L., Subspace-Based Signal Analysis Using Singular Value Decomposition. *Proceedings of the IEEE*, Vol. 81, 1993, 1277-1308.

[24] Oshman, Y. and Bar-Itzhack, I.Y., Square Root Filtering via Covariance and Information Eigenfactors, *Automatica*, Vol. 22, No.5, 1986, 599-604.

[25] Oshman, Y. Gain-Free Square Root Information Filtering Using the Spectral Decomposition, *J. Guid., Contr., and Dynamics*, Vol. 12, No.5, 1989, 681-690.

[26] Lu, M., Qiao, X. and Chen, G., Parallel Computation of the Modified Extended Kalman Filter, *Int. J. Computer Math.*, Vol. 45, 1992, 69-87.

[27] Wang, L., Libert, G. and Manneback, P., Kalman Filter Algorithm Based on Singular Value Decomposition, *Proc. of the 31st Conf. on Decision and Control*, 1992, 1224-1229.

[28] Zhang, Y.M., Dai, G.Z., Zhang, H.C. and Li, Q.G., A SVD-Based Extended Kalman Filter and Applications to Aircraft Flight State and Parameter Estimation, *Proc. of 1994 American Control Conf.*, Baltimore, MD, 1994, 1809-1813.

[29] Zhang, Y.M., Li, Q.G., Dai, G.Z. and Zhang, H.C., A New Recursive Identification Method Based on Singular Value Decomposition, *J. Control Theory and Application*, Vol. 12, No.2, 1995, 224-229.

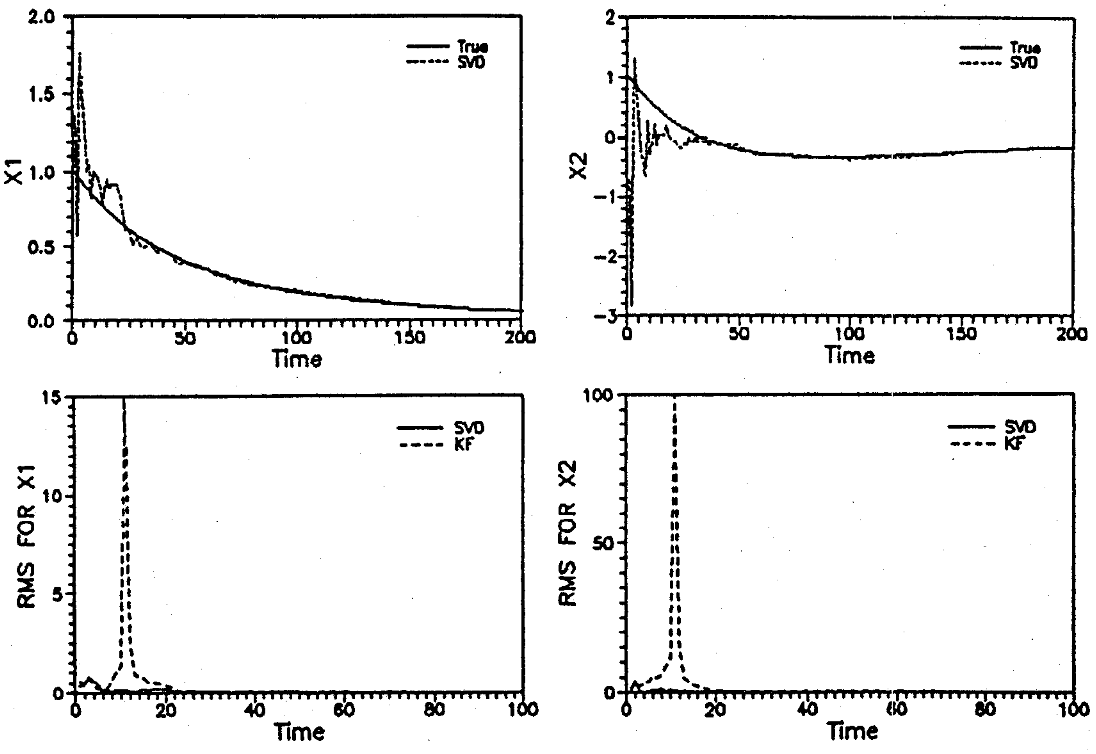

Figure 1. The comparison of SVD-based Kalman filter with Kalman filter

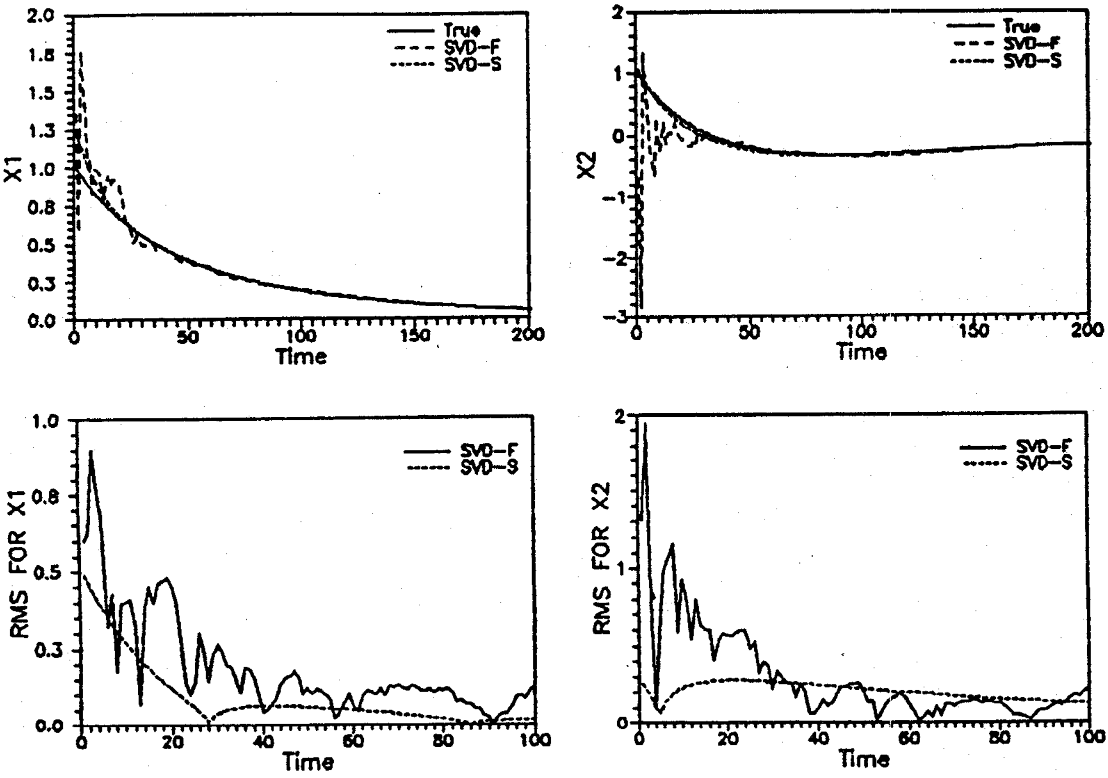

Figure 2. The comparison of SVD-based smoother with SVD-based Kalman filter