ROBUST RECURSIVE BI-ITERATION SINGULAR VALUE DECOMPOSITION (SVD) FOR SUBSPACE TRACKING AND ADAPTIVE FILTERING

Y. Wen, S. C. Chan, and *K. L. Ho*

Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong
email: ywen@eee.hku.hk, scchan@eee.hku.hk, klho@eee.hku.hk

ABSTRACT

The recursive bi-iteration singular value decomposition (Bi-SVD), proposed by Strobach [1], is en efficient and well-structured algorithm for performing subspace tracking. Unfortunately, its performance under impulse noise environment degrades substantially. In this paper, a new robust-statistics-based bi-iteration SVD algorithm (robust Bi-SVD) is proposed. Simulation results show that the proposed algorithm offers significantly improved robustness against impulse noise than the conventional Bi-SVD algorithm with slight increase in arithmetic complexity. For nominal Gaussian noise, the two algorithms have similar performance.

1. INTRODUCTION

Subspace decomposition is a very valuable tool in signal processing, digital communications, and array signal processing. The singular value decomposition, SVD, is a known to be a good method for such computation because of its exact numerical properties. However, direct computation of the SVD is an iterative process and its complexity is very high in real-time applications. In many practical applications, only a sub-set of the singular vectors and singular values of the SVD have to be computed from a recursively updated correlation matrix. These properties can be utilized to obtain efficient algorithms for tracking the sub-set of singular vectors required. A number of recursive eigen-decomposition algorithms have also been proposed for recursive subspace tracking [1-4]. In particular, the Bi-Iteration Singular Value Decomposition (Bi-SVD) proposed by Stobach [1], which is based on an extension of Bauer's classical bi-iteration [16] to sequential SVD updating, has a comparable performance to the well-known reference algorithm of Karasal [4], but the Bi-SVD has a regular structure and much lower complexity. Unfortunately, when it is applied to subspace tracking applications where impulsive noise sporadically intrudes the system, say in a communication channel, its performance degrade substantially. The deeper reason for this fragility is that the conventional autocorrelation matrix estimate: $R_{\hat{x}\hat{x}}(n)=\sum_{i=1}^{n}\lambda^{n-i}\hat{x}\hat{x}^{T}$, where λ is the forgetting factor and $\hat{x}$ is the input signal vector, is not a robust estimate of the underlying autocorrelation $R_{xx}=E[xx^{T}]$, if x is corrupted by the impulse noise to form $\hat{x}$ [11-13]. This problem has been studied in robust statistics [14] and the minimum volume ellipsoid (MVE) or other robust estimators should be used. However, the computational complexity is usually prohibitive for real-time applications. In recursive subspace estimation, a rough prior knowledge of the subspace estimate is available from previous iterations. Therefore, it is easier to detect whether the incoming signal vector is potentially corrupted by impulse noise or not. This idea happens to coincide with the M-estimators or Maximum likelihood like estimators of the correlation matrix. It is therefore not surprising that the Bi-SVD algorithm and other RLS-based subspace tracking algorithms [11-13] are sensitive to impulsive or non-Gaussian noise, as we shall see later from the simulation results.

In this paper, a new robust Bi-SVD recursive subspace tracking algorithm using a robust statistics approach similar to [5,14] is proposed. In particular, a method for detecting an impulse-corrupted data vector is developed based on the statistics of the orthogonal innovation vector, which is the complement vector of the orthogonal projection of data vector on "old " singular subspace. Data samples, which are potentially corrupted by impulse noise, are then replaced by appropriate estimate from previously "cleaned" data samples. The layout of the paper is as follows: Section 2 is a brief description of the Bi-SVD subspace tracking problem. The proposed robust Bi-SVD subspace tracking algorithm is introduced in Section 3. Simulation results and comparison with the conventional method are presented in Section 4. And finally, conclusions are drawn in Section 5.

2. BI-ITERATION SVD ALGORITHM

The Bi-SVD subspace tracking algorithm [1] computes the r dominant singular values and corresponding right singular vectors from the following growing data matrix $X(t)$:

$$X(t)=\begin{bmatrix}(1-\alpha)^{1/2}x^{T}(t)\\ \alpha^{1/2}X(t-1)\end{bmatrix} \tag{1}$$

where $X(t-1)$ is the old data matrix at time t-1, $x(t)=[x(t),x(t-1),...,x(t-N-1)]^{T}$ is the newly incoming data vector of dimension N at time t, $x(t)$ $(t=1,2,3,...)$ is the received data samples, and $0\le\alpha\le1$ is a *non-negative exponential forgetting* factor. The classical bi-iteration SVD algorithm applied to the $L\times N$ real data matrix X is shown in the following:

Classical Bi-Iteration SVD Algorithm

```
Initialize $Q_A(0)=\begin{bmatrix}I_r\\0\end{bmatrix}$
FOR $\mu=1,2,....$ DO (Until Converge)
    $B(\mu)=XQ_A(\mu-1)$,
    $B(\mu)=Q_B(\mu)R_B(\mu)$, ( $L\times r$ $QR$ -Factorization)
    $A(\mu)=X^{T}Q_B(\mu)$,
    $A(\mu)=Q_A(\mu)R_A(\mu)$, ( $N\times r$ $QR$ -Factorization).
```

It has been proved that the orthonormal columns of the $L\times r$ matrix $Q_B(\mu)$ will converge to the r dominant left singular vectors, and the orthonormal columns of the $N\times r$ matrix $Q_A(\mu)$ will converge to the r dominant right singular vectors of the SVD of the data matrix X. At the same time, both triangular matrices $R_A(\mu)$ and $R_B(\mu)$ will converge to the $r\times r$ diagonal matrix of dominant singular values of X. Having replaced the index μ with the discrete time index t and made a *consistent approximation* of,

$$\hat{X}(t)=Q_B(t)R_B(t)Q_A^{T}(t-1) \tag{2}$$

Strobach [1] extends this classical bi-iteration SVD to a sequential square-root type bi-iteration SVD subspace-updating algorithm as follows:

0-7803-7761-3/03/$17.00 ©2003 IEEE

```
Bi-Iteration SVD Subspace Tracking
Initialize $Q_A(t-1)=\begin{bmatrix} I_r \\ 0 \end{bmatrix}$; $R_B(t-1)=\theta_A(t-1)=I_r$;
        $R_A(t-1)=[0]$;   $r$ is rank.
FOR $t=1,2,....$ DO
  Input $x(t)$
  $h(t)=Q_A^T(t-1)x(t)$, $x_\perp(t)=x(t)-Q_A(t-1)h(t)$,
  $e_x(t)=x_\perp^T(t)x_\perp(t)$, $\bar{x}_\perp(t)=e_x^{-1/2}(t)x_\perp(t)$,
  $H(t)=R_B(t-1)\theta_A(t-1)$,
  $\begin{bmatrix} R_B(t) \\ --- \\ 0\cdots0 \end{bmatrix}=G_B(t)\begin{bmatrix} (1-\alpha)^{1/2}h^T(t) \\ ------ \\ \alpha^{1/2}H(t) \end{bmatrix}$,
  $h_R^T(t)R_B(t)=h^T(t) \xrightarrow{\text{back substitution}} h_R^T(t)$,
  $H_R(t)R_B(t)=H(t) \xrightarrow{\text{back substitution}} H_R(t)$,
  $\begin{bmatrix} R_A(t) \\ --- \\ 0\cdots0 \end{bmatrix}=G_A(t)\begin{bmatrix} \alpha R_A(t-1)H_R(t)+(1-\alpha)h(t)h_R^T(t) \\ ------------------ \\ (1-\alpha)e_x^{1/2}(t)h_R^T(t) \end{bmatrix}$
  $G_A^T(t)=\left[\begin{array}{c|c} \theta_A(t) & \begin{matrix} * \\ \vdots \\ * \end{matrix} \\ \hline f^T(t) & * \end{array}\right] \xrightarrow{\text{extract}} \theta_A(t); f(t)$,
  $Q_A(t)=Q_A(t-1)\theta_A(t)+\bar{x}_\perp(t)f^T(t)$,
END
```

TABLE 1.

The compressed data vector $h(t)$ of dimension r is actually the *projection* of the input data vector $x(t)$ onto the right singular subspace basis $Q_A^T(t-1)$; $x_\perp(t)$ is the complement of this orthogonal *projection*; $e_x(t)$ the squared norm of the complement vector $x_\perp(t)$; and $\bar{x}_\perp(t)$ is the normalized complement vector which acts as the *innovation* for subspace updating. The $r\times r$ cosine matrix $\theta_A(t)=Q_A^T(t-1)Q_A(t)$ describes the *distance* between consecutive subspaces, while $G_B(t)$ and $G_A(t)$ represent a sequence of orthonormal Givens plane rotations, which are determined in such a way that the rotated old triangular matrix on its right hand side is transformed into a strictly upper-right triangular matrix on its left hand side. $f(t)$ is the *compressed innovation* vector defined as $f(t)=Q_A^T(t)\bar{x}_\perp(t)$. $H(t)$, $h_R^T(t)$, $H_R(t)$ are vectors and matrixes defined respectively as $H(t)=R_B(t-1)\theta_A(t-1)$, $h_R^T(t)=h^T(t)R_B^{-1}(t)$, and $H_R(t)=H(t)R_B^{-1}(t)$. Like the PAST algorithm, the overall computation can also be viewed as a mean of minimizing $e_x(t)$, the squared norm of the complement vector $x_\perp(t)$. As mentioned earlier, the performance of this algorithm is very sensitive to impulse noise, because the *squared norm* of $x_\perp(t)$ is minimized, which is not a robust estimator. In fact, if $x(t)$ is corrupted by impulse noise, $h(t)$, $x_\perp(t)$, $R_B(t)$, $h_R^T(t)$, $H_R(t)$, $R_A(t)$, $G_A(t)$, $\theta_A(t)$, and $f(t)$ will all be affected in turn. The new subspace estimate will hence be perturbed to a point which is far away from the true subspace, even though the impulse noise power is merely 20 dB to 25 dB higher than nominal Gaussian background noise. More importantly, the corrupted matrices, $R_B(t)$, will be used to compute the new $R_B(t)$'s, which takes the Bi-SVD algorithm many iterations to recover, especially when α is close to one.

3. ROBUST BI-SVD SUBSPACE TRACKING

We now consider the proposed robust statistics-based Bi-SVD algorithm. First of all, we note that when $x(t)$ is corrupted by one or more impulses, $e_x(t)$, which is the squared norm of complement vector $x_\perp(t)$, in the Bi-SVD algorithm mentioned above will become very large. Instead of using the squared norm, an M-estimator $E[e_x(t)]=E[q(e_x(t))\cdot e_x(t)]$ can be used, where $q(e_x(t))$ is a weighting factor which de-emphasis error norm with exceptional large value. For simplicity, we consider the modified Huber function, where $q(e_x(t))$ is equal to 0 and 1 respectively when $e_x(t)$ is greater than or smaller than a threshold T, which is to be estimated continuously. Though the exact distribution of the $e_x(t)$ is unknown, it is assumed for simplicity to be Gaussian distributed but corrupted by additive impulse noise to simplify the detection of the impulses (note also that $e_x(t)$ is always positive). It then follows that the probability for $|\Delta e_\mu(t)|=|e_x(t)-\hat{\mu}(t)|$ to be greater than a given threshold $T(t)$ is

$$p_T=P_r\{|\Delta e_\mu(t)|>T(t)\} = erfc\left(T(t)/\hat{\sigma}(t)\right), \qquad (3)$$

where $erfc(r)=(2/\sqrt{\pi})\int_r^\infty e^{-x^2}dx$ is the complementary error function. $\hat{\mu}(t)$ and $\hat{\sigma}(t)$ are the estimated mean and standard deviation of the squared norm $e_x(t)$ of the "impulse free" $x_\perp(t)$. Using different threshold parameters $T(t)$, we can detect whether the incoming vector is potentially corrupted by impulse noise with different degrees of confidence. In this work, p_T is chosen to be 0.05 so that we have 95% confidence in saying that the current data vector $x(t)$ is corrupted by impulse noise. The selection of this threshold is a tradeoff between impulse suppression and signal distortion. The larger the threshold, the smaller will be the signal distortion, however, at the expense of less immunity to impulse noise. In practice, a threshold value corresponding to a confidence of 93% to 97% works well. For 95% confidence, the corresponding threshold parameter $T(t)$ is determined to be $T(t)=1.96\cdot\hat{\sigma}(t)$. A commonly used estimate for $\hat{\sigma}^2(t)$, and $\hat{\mu}(t)$ are $\hat{\sigma}^2(t)=\lambda_\sigma\hat{\sigma}^2(t-1)+(1-\lambda_\sigma)(\Delta e_\mu(t))^2$, and $\hat{\mu}(t)=\lambda_\mu\hat{\mu}(t-1)+(1-\lambda_\mu)e_x(t)$, respectively, where λ_μ and λ_σ are some forgetting factors. It is, however, not robust to impulse noise. In fact, a single impulse with large amplitude can substantially increase the value of $\hat{\sigma}(t)$ and $\hat{\mu}(t)$, and hence the values of $T(t)$. Following [6], we employ the following robust estimates for $\hat{\sigma}^2(t)$ and $\hat{\mu}(t)$

$$\hat{\sigma}^2(t)=\lambda_\sigma\hat{\sigma}^2(t-1)+1.483\left(1+\frac{5}{N_w-1}\right)(1-\lambda_\sigma)\operatorname{med}\left(A((\Delta e_\mu(t))^2)\right)$$

$$\text{and}\quad \hat{\mu}(t)=\lambda_\mu\hat{\mu}(t-1)+(1-\lambda_\mu)\operatorname{med}\left(A(e_x(t))\right), \qquad (4)$$

where $A(x(t))=\{x(t),\cdots,x(t-N_w+1)\}$, N_w is the length of the estimation window, and med(.) is the median operation. λ_μ and λ_σ are the forgetting factors. It can be seen from the above discussion that the arithmetic complexity of the proposed robust Bi-SVD algorithm is very close to that of the conventional Bi-SVD algorithm. For very large values of N_w, the complexity for performing the medium filter can be significantly reduced by computing the pseudo median [17], instead of the median. The pseudo median is an approximation to the median with much lower complexity and can be efficiently implemented in a pipeline structure. In practice, N_w is limited to 5 to 11, therefore the increase in complexity is quite acceptable. Our robust Bi-SVD algorithm updates $T(t)=1.96\cdot\hat{\sigma}(t)$, $\hat{\sigma}^2(t)$ and $\hat{\mu}(t)$ at each iteration. If $|\Delta e_\mu(t)|>T(t)$, the incoming received data sample $x(t)$ in the new date vector $\boldsymbol{x}(t)=[x(t),x(t-1),...,x(t-N-1)]^T$ is treated as a impulse-corrupted data sample. To prevent it from entering the subspace updating iteration, $x(t)$ has to be replaced by certain estimate such as the linear predictor from previous "impulse free" samples. For simplicity, the robust mean estimate of the received data sample $\hat{\mu}_s(t)$ is used:

$$\hat{\mu}_s(t)=\beta_s\hat{\mu}_s(t-1)+(1-\beta_s)med(A(x(t))), \qquad (5)$$

where β_s is its forgetting factor. This scheme is shown to be very effective in blocking the impulse noise from entering the Bi-SVD algorithm. Finally, we obtain the robust Bi-SVD subspace tracking algorithm in TABLE 2.

4. SIMULATION RESULTS

The performance of the proposed robust Bi-SVD algorithm is evaluated under the ε-contamination noise model, and it is compared to the conventional Bi-SVD algorithm. The probability density function (pdf) of the ε-contamination noise model is

$$f=(1-\varepsilon)N(0,\sigma^2)+\varepsilon N(0,\sigma_I{}^2), \qquad (6)$$

where $0\le\varepsilon\le1$, $N(0,\sigma^2)$ is the pdf of the Gaussian background noise, $N(0,\sigma_I{}^2)$ is the pdf of the impulse noise. In the simulation, the impulse noise occurrence probability ε was set to 0.1, with $10\cdot\log_{10}\frac{\sigma_I{}^2}{\sigma^2}=25dB$. The two algorithms are applied to track the signal subspace of the received data sequence, which is the superposition of two temporary monochromatic sources and noise. After each subspace update, adaptive subspace filtering is implemented to reconstruct the two source signals from the corrupting noise. The adaptive subspace filtering performed is:

$$\hat{\boldsymbol{s}}(t)=\boldsymbol{Q}_A(t)\boldsymbol{Q}_A^T(t)\boldsymbol{x}(t), \qquad (7)$$

and the actual sample $\hat{s}(t)$ of the reconstructed signal sequence is extracted via the bottom pinning of the vector $\boldsymbol{s}_\Sigma(t)$, which is the sum of successively shifted vector $\hat{\boldsymbol{s}}(t)$ as:

$$\boldsymbol{s}_\Sigma(t)=\boldsymbol{D}\boldsymbol{s}_\Sigma(t-1)+\frac{1}{N}\hat{\boldsymbol{s}}(t),$$

$$\boldsymbol{D}=\begin{bmatrix}0\cdots0 & 0\\ & 0\\ \boldsymbol{I}_{N-1} & \vdots\\ & 0\end{bmatrix},\quad \hat{s}(t)=[0\cdots0,1]\boldsymbol{s}_\Sigma(t). \qquad (8)$$

Two temporary monochromatic source signals are sinusoidal sequences with normalized frequencies of $\omega_1=10^o$, and $\omega_2=12^o$, respectively. Each source signal is $-4.88\,dB$ below the Gaussian background noise. The parameters for the two algorithms are: order of data vector: $N=181$, forgetting factor: $\alpha=0.993$, rank of signal subspace: $r=4$. For the robust Bi-SVD, the length of the median filter N_w is set to 9, and the forgetting factors λ_μ, λ_σ, and β_s are all set equal to 0.97. The initial value $\hat{\mu}_s(0)$ is chosen to be zero. Both $\hat{\sigma}^2(0)$ and $\hat{\mu}(0)$ are set to 10, a relatively large number to their normal value, to initialize system adaptation. Fig.1 shows the test data: Fig.1 (a) is the first sinusoidal sequence with normalized frequency $\omega_1=10^o$. Fig.1 (b) is the second sinusoidal sequence with normalized frequency $\omega_2=12^o$. Fig.1 (c) is the transmitted signal constituting of the sum of these two sinusoidal sequences. Fig.1 (d) is the channel noise, where impulse noise intrudes the channel during the time interval from the 1000th sample to the 1500th sample. Fig.1 (e) shows the received samples, which is the sum of the transmitted signal and channel noise. Fig.2 depicts the signal reconstruction process using Bi-SVD based adaptive subspace filtering. From Fig.2 (b), we can see that the reconstructed samples are severely distorted by the impulse noise in the interval between the 1000th sample and the 1500th sample. Moreover, the reconstruction error during this interval has a shape resembling the impulsive noise, which suggests that they are malign errors. In addition, even though the impulsive noise passes away after the 1500th sample, the reconstruction error couldn't converge to the normal level until the 2000th sample. Fig.3 shows the mean maximum principal angle (a measurement of "distance") between the Bi-SVD subspace estimate and the true subspace derived from the batch eigendecomposition of noise-free data covariance matrix. This mean maximum principal angle was obtained by averaging over 100 independent Monte Carlo trials. It can be observed in Fig.3 that the Bi-SVD subspace estimate is significantly affected by the impulse noise (very large maximum principle angle) and deviates far away from its true value. It is unable to re-converge to true signal subspace (small maximum principle angle) until 2000th sample. In contrast, Fig.4 shows that reconstruction error of the proposed robust Bi-SVD is much smaller than that of the Bi-SVD algorithm, and it is similar to the transmitted signal, despite the presence of the impulse noise. Besides, the reconstruction error re-converged to the normal level almost right after the impulse noise was turned off. Fig.5 substantiates the robustness of the proposed robust Bi-SVD algorithm, where it can be seen that the subspace estimate remains largely unaffected and is always close to the true subspace in the presence of the impulses.

5. CONCLUSION

A new robust-statistics-based Bi-SVD subspace tracking algorithm in impulse noise environment is presented. Simulation results using adaptive subspace-filtering show that the proposed algorithm offers improved robustness than the conventional Bi-SVD algorithm under contaminated Gaussian noise environment. The computational complexity of the robust Bi-SVD algorithm is slightly higher than that of the conventional Bi-SVD algorithm

6. REFERENCES

[1] P. Strobach, "Bi-Iteration SVD Subspace Tracking Algorithms", *IEEE Trans SP*, vol.45, no.5, pp.1222-1240, May 1997.

[2] P. Strobach, "Low-Rank Adaptive Filters", *IEEE Trans SP*, vol.44, no.12, pp.2932-2947, December 1996.

[3] P. Strobach, "Bi-Iteration Recursive Instrumental Variable Subspace Tracking and Adaptive Filtering", *IEEE Trans SP*, vol.46, no.10, pp.2708-2725, October 1998.

[4] I. Karasalo, "Estimating The Covariance Matrix By Signal Subspace Averaging", *IEEE Trans Acoustl Speech, Signal Processing*, ASSP-34, pp.8-12, Feb. 1986.

[5] Y. X. Zou, S. C. Chan, and T. S. Ng, "A Robust M-Estimate Adaptive Filter for Impulsive Noise Suppression", *Proceedings of ICASSP'99*, Vol.4, Phoenix, Arizona, USA, March 1999.

[6] Y. X. Zou, S. C. Chan and T. S. Ng, "A Recursive M-estimate Algorithm for Robust Adaptive Filtering in Impulse Noise," *IEEE Sig. Proc. Letter*, Nov. 2000, pp. 324 -326.

[7] Y. X. Zou, S. C. Chan and T. S. Ng, "Fast Least Mean *M*-estimate Algorithms for Robust Adaptive filtering in impulse noise," *IEEE Circuits and Systems II*, Dec. 2000, pp. 1564-1569.

[8] Y. X. Zou, S. C. Chan and T. S. Ng, "Robust M-Estimate Adaptive Filtering," *IEE Proc. Signal Processing*, vol.148, no.4, pp. 289-294, Aug. 2001.

[9] Y. X. Zou, S. C. Chan, and T. S. Ng, "A Robust Statistics Based Adaptive Lattice-Ladder Filter In Impulsive Noise, " in *Proc. IEEE ISCAS'2000*, Geneva.

[10] Y. X. Zou, S. C. Chan and T. S. Ng, "Fast Least Mean M-Estimate Algorithms for Robust Adaptive Filtering in Impulse Noise," in *Proc. EUSIPCO'2000*, Tampere, Finland.

[11] Y.Wen, S.C. Chan, and K.L. Ho, "Robust Subspace Tracking In Impulsive Noise" in *Proc. ICC'2001*, Helsinki, Finland.

[12] Y.Wen, S.C. Chan, and K.L. Ho, "Robust Subspace Tracking Base Blind Channel Identification In Impulsive Noise Environment" in *Proc. EUSIPCO2002*, Toulouse, France.

[13] Y.Wen, S.C. Chan, and K.L. Ho, "A Robust Subspace Tracking Algorithm For Subspace-Based Blind Multiuser Detection In Impulsive Noise" in *Proc. DSP2002*, Santorini, Greece.

[14] P. J. Huber, "Robust statistics", John Wiley, New York, 1981.

[15] G. H. Golub, C. F. Van Loan, "Matrix Computations", The Johns Hopkins Uni. Press, Baltimore and London, 1996.

[16] F. L. Bauer, "Das Verfahren der Treppeniteration und verwandte Verfahren zur L"osung algebraischer Eigenwertprobleme" Z. Angew. Math. Phys., vol. 8, pp. 214–235, 1957.

[17] W. K. Pratt, Digital Image Processing, 2nd Edition. John Wiley & Sons, Inc. 1991.

```
Bi-Iteration SVD Subspace Tracking
Initialize $Q_A(t-1)=\begin{bmatrix} I_r \\ 0 \end{bmatrix}$; $R_B(t-1)=\theta_A(t-1)=I_r$;
        $R_A(t-1)=[0]$; $\hat{\sigma}^2(0)$, $\hat{\mu}(0)$, and $\hat{\mu}_s(0)$; $r$ is rank.
FOR $t=1,2,....$ DO
    Input $x(t)$
    $h(t)=Q_A^T(t-1)x(t)$, $x_\perp(t)=x(t)-Q_A(t-1)h(t)$,
    $e_x(t)=x_\perp^T(t)x_\perp(t)$,
    IF $|\Delta e_\mu(t)|>T(t)$ THEN
            $x(t)=\hat{\mu}_s(t-1)$, $h(t)=Q_A^T(t-1)x(t)$,
            $x_\perp(t)=x(t)-Q_A(t-1)h(t)$, $e_x(t)=x_\perp^T(t)x_\perp(t)$,
    $\bar{x}_\perp(t)=e_x^{-1/2}(t)x_\perp(t)$, $H(t)=R_B(t-1)\theta_A(t-1)$,
    $\begin{bmatrix} R_B(t) \\ --- \\ 0\cdots0 \end{bmatrix}=G_B(t)\begin{bmatrix} (1-\alpha)^{1/2}h^T(t) \\ ------ \\ \alpha^{1/2}H(t) \end{bmatrix}$,
    $h_R^T(t)R_B(t)=h^T(t) \xrightarrow{back\ substitution} h_R^T(t)$,
    $H_R(t)R_B(t)=H(t) \xrightarrow{back\ substitution} H_R(t)$,
    $\begin{bmatrix} R_A(t) \\ --- \\ 0\cdots0 \end{bmatrix}=G_A(t)\begin{bmatrix} \alpha R_A(t-1)H_R(t)+(1-\alpha)h(t)h_R^T(t) \\ ------------------ \\ (1-\alpha)e_x^{1/2}(t)h_R^T(t) \end{bmatrix}$
    $G_A^T(t)=\left[\begin{array}{c|c} \theta_A(t) & \begin{matrix} * \\ \vdots \\ * \end{matrix} \\ \hline f^T(t) & * \end{array}\right] \xrightarrow{extract} \theta_A(t); f(t)$,
    $Q_A(t)=Q_A(t-1)\theta_A(t)+\bar{x}_\perp(t)f^T(t)$,
    update $\hat{\sigma}^2(i)$, $\hat{\mu}(i)$ and $\hat{\mu}_s(0)$ using(4)(5), $T(i)=1.96\cdot\hat{\sigma}(i)$,
END
```

TABLE 2.

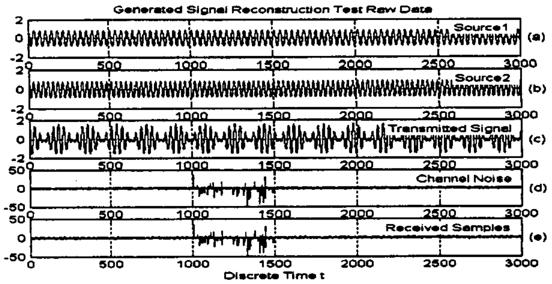

Fig. 1

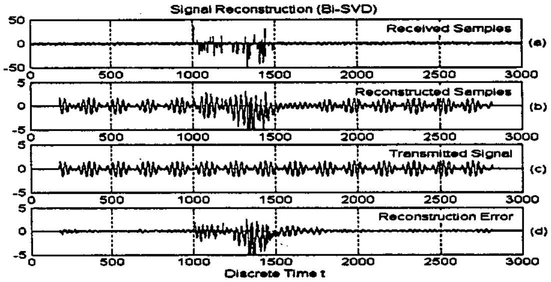

Fig. 2

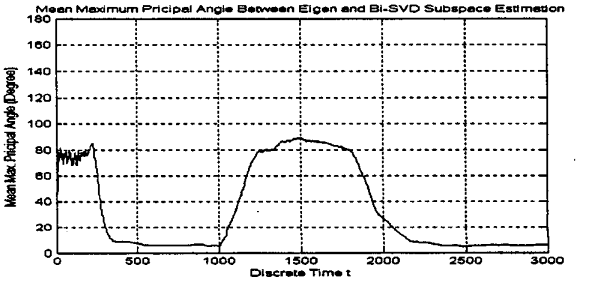

Fig. 3

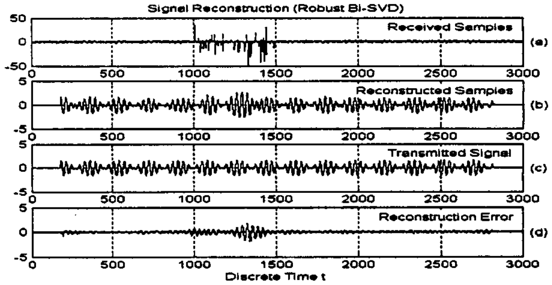

Fig. 4

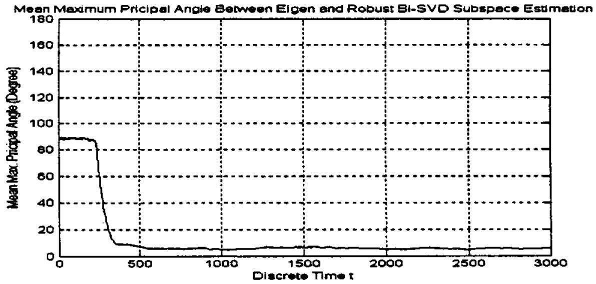

Fig. 5