Journal of VLSI Signal Processing, 14, 75–92 (1996)

Subspace Tracking with Adaptive Threshold Rank Estimation

ALEKSANDAR KAVČIĆ

Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

BIN YANG

Department of Electrical Engineering, Ruhr University Bochum, 44780 Bochum, Germany

Received October 2, 1995. Revised April 2, 1996.

Editors:

Abstract. In frequency and direction of arrival (DOA) tracking problems, singular value decomposition (SVD) can be used to track the signal subspace. Typically, for a problem size n, only a few, say r dominant eigencomponents need to be tracked, where $r \ll n$. In this paper we show how to modify the Jacobi-type SVD to track only the r-dimensional signal subspace by forcing the $(n-r)$-dimensional noise subspace to be spherical. Thereby, the computational complexity is brought down from $O(n^2)$ to $O(nr)$ per update. In addition to tracking the subspace itself, we demonstrate how to exploit the structure of the Jacobi-type SVD to estimate the signal subspace dimension via a simple adaptive threshold comparison technique. Most available computationally efficient subspace tracking algorithms rely on off-line estimation of the signal subspace dimension, which acts as a bottleneck in real-time parallel implementations. The noise averaged Jacobi-type SVD updating algorithm presented in this paper is capable of simultaneously tracking the signal subspace and its dimension, while preserving both the low computational cost of $O(nr)$ and the parallel structure of the method, as demonstrated in a systolic implementation. Furthermore, the algorithm tracks all signal singular values. Their squares are estimates of the powers in the orthogonal modes of the signal. Thus, applications of the algorithm are not limited to only DOA and frequency tracking where information about the powers of signal components is not exploited.

Keywords: eigenvalue decomposition, singular value decomposition, QR-decomposition, Jacobi rotation, Jacobi-type singular value decomposition, subspace tracking, rank estimation

1. Introduction

Subspace based high resolution methods have been applied successfully to direction-of-arrival (DOA) estimation and frequency retrieval. Typical examples are the Pisarenko method [1], the multiple signal classification (MUSIC) algorithm [2], [3], the minimum-norm method [4], the ESPRIT estimator [5], and the weighted subspace fitting (WSF) algorithm [6]. In off-line applications, a block of sensor array or time series data is collected to form a sample correlation matrix whose eigenvalue decomposition (ED) or, for better numerical accuracy, the singular value decomposition (SVD) of the data matrix is computed. The eigenvalues (squares of the corresponding singular values) are used to estimate the number of signals and the eigenvectors corresponding to the right-hand side singular vectors are involved in estimating the DOAs or frequencies.

This approach, however, is unsuitable for online tracking because ED and SVD are very time consuming. For example, ED of an $n \times n$ correlation matrix requires $O(n^3)$ operations. Therefore, adaptive algorithms for computing the signal and/or noise subspace recursively have recently attracted much attention [7]-[19]. Most of these algorithms share the common feature that they update only a few, say r ($r < n$), desired eigencomponents instead of the full eigenstructure for each new arriving sample vector. Hence, they of-

fer a lower computational complexity than batch ED or SVD.

In this paper, we present a novel method for tracking the signal subspace. Starting with the Jacobi-type SVD updating algorithm proposed by Moonen, Van Dooren, and Vandewalle [12], we apply the noise averaging approach to force a spherical noise subspace. This allows us to deflate the problem and to update the r-dimensional signal subspace with only $O(nr)$ operations per update. The basic computational steps are interlaced QR-decomposition and Jacobi rotations as well as re-averaging of the noise singular values. Notice that the full SVD updating by Moonen et al. demands $O(n^2)$ operations.

A common weakness of most subspace tracking algorithms is the assumption that the dimension of the signal subspace is given and fixed. In many applications, the number of uncorrelated signals, i.e. the rank of the signal part of the correlation matrix, is unknown and/or time varying. In this case, we need both rank and subspace tracking. In this paper, we shall address the problem of rank estimation too. We propose two different procedures for simultaneous rank and subspace tracking, one based on exploiting the information theoretic criteria and the other based on thresholding the averaged noise singular value. The operation count for both rank and subspace tracking remains $O(nr)$.

Moreover, we show that the algorithm is capable of highly concurrent computations. It suits for a systolic implementation on an array with $r^2/2 + O(r)$ processing elements providing an update in $O(n)$ time. Hence, this algorithm combines uniquely the low computational cost with simultaneous rank estimation and parallel implementation.

This paper is organized as follows. In section 2, we briefly review the underlying signal model and its eigenstructure properties. Section 3 gives a short description of the Jacobi-type noise averaged SVD (NASVD) updating updating algorithm [20], [21]. In section 4 we discuss extensions of the NASVD algorithm to allow for simultaneous subspace dimension tracking. There we discuss an adaptation of information theoretic criteria for purposes of noise averaged subspace tracking and we present a new alternative adaptive threshold comparison criterion. Section 5 contains a discussion on the optimal threshold for the threshold comparison dimension tracking criterion. We address the issue of systolic implementation briefly in section 6. Results of numerical experiments, demonstrating the subspace and rank tracking capabilities of the Jacobi-type NASVD updating algorithm are shown in section 7.

The following notations are used in this paper. Boldface and underlined characters are used to represent matrices and column vectors, respectively. The superscript $^{\mathrm{H}}$ denotes the Hermitian transposition. $\|\cdot\|$ represents the 2-norm of matrices and vectors. The (i,j)-th element of the matrix $\mathbf{A}$ is denoted by $\mathrm{A}\{i,j\}$.

2. The Signal Model

In DOA and frequency estimation, the problem of tracking the ED of a time varying signal correlation matrix

$$\mathbf{C}_k = \mathrm{E}\left[\underline{x}_k\,\underline{x}_k^{\mathrm{H}}\right] = \mathbf{S}_k\mathbf{C}_k^{\mathrm{S}}\mathbf{S}_k^{\mathrm{H}} + \lambda_k^{\mathrm{N}}\mathbf{I} = \mathbf{W}_k\mathbf{\Lambda}_k\mathbf{W}_k^{\mathrm{H}} \tag{1}$$

arises [7], [22]. Here, k $(0<k<\infty)$ represents a time index. The signal vector $\underline{x}_k$ is a snapshot of an n-element sensor array in case of DOA tracking or the output of an n-tap delay line in case of frequency tracking. We assume $\underline{x}_k$ to be composed of r $(r<n)$ signal waves impinging an antenna array or r incoherent complex sinusoids corrupted by additive white noise. $\mathbf{S}_k$ is an $n\times r$ full rank matrix which is determined by the DOAs or frequencies. $\mathbf{C}_k^{\mathrm{S}}$ is an $r\times r$ full rank correlation matrix of the r signals. λ_k^{N} denotes the white noise variance. Hence, the n eigenvalues of $\mathbf{C}_k$ are given by

$$\begin{aligned}\lambda_k^{\mathrm{S}}(1) \geq \lambda_k^{\mathrm{S}}(2) \geq \cdots \geq \lambda_k^{\mathrm{S}}(r) > \\ \lambda_k^{\mathrm{N}}(1) = \lambda_k^{\mathrm{N}}(2) = \cdots = \lambda_k^{\mathrm{N}}(n-r) = \lambda_k^{\mathrm{N}}.\end{aligned} \tag{2}$$

The r largest eigenvalues marked with the superscript $^{\mathrm{S}}$ are referred to as signal eigenvalues, while the remaining eigenvalues have the superscript $^{\mathrm{N}}$ marking them as noise eigenvalues. The eigenvalue matrix $\mathbf{\Lambda}_k$ can thus be represented as $\mathbf{\Lambda}_k = \mathrm{diag}\left(\mathbf{\Lambda}_k^{\mathrm{S}}, \mathbf{\Lambda}_k^{\mathrm{N}}\right)$, where $\mathbf{\Lambda}_k^{\mathrm{S}} = \mathrm{diag}\left(\lambda_k^{\mathrm{S}}(1), \ldots, \lambda_k^{\mathrm{S}}(r)\right)$ and $\mathbf{\Lambda}_k^{\mathrm{N}} = \lambda_k^{\mathrm{N}}\mathbf{I}_{n-r}$. Consistent with this notation, we break the eigenvector matrix as $\mathbf{W}_k = \left[\begin{array}{cc}\mathbf{W}_k^{\mathrm{S}} & \mathbf{W}_k^{\mathrm{N}}\end{array}\right]$. Here, the r

columns of $\mathbf{W}_k^{\mathrm{S}}$ are the signal eigenvectors (corresponding to the signal eigenvalues) and the $n-r$ columns of $\mathbf{W}_k^{\mathrm{N}}$ are the noise eigenvectors.

Since in real applications the correlation matrix is not known, one needs to estimate it from observed data. We estimate the correlation matrix by exponentially weighting rank-one sample vector outer products

$$\hat{\mathbf{C}}_k = \sum_{i=1}^{k} \beta^{k-i} \underline{x}_i \underline{x}_i^{\mathrm{H}} = \beta \hat{\mathbf{C}}_{k-1} + \underline{x}_k \underline{x}_k^{\mathrm{H}}, \tag{3}$$

where β ($0<\beta<1$) is the exponential forgetting factor. By performing the ED of $\hat{\mathbf{C}}_k$, we obtain estimates for the signal and noise eigenvector matrices $\mathbf{V}_k^{\mathrm{S}}=\hat{\mathbf{W}}_k^{\mathrm{S}}$ and $\mathbf{V}_k^{\mathrm{N}}=\hat{\mathbf{W}}_k^{\mathrm{N}}$. Thereby, our initial ED tracking problem becomes tracking the ED of the estimated correlation matrix $\hat{\mathbf{C}}_k$ in (3).

Notice that the rank-one update in (3) can also be formulated as $\hat{\mathbf{C}}_k = \mathbf{A}_k^{\mathrm{H}} \mathbf{A}_k$ with the recursive formula for the weighted data matrix

$$\mathbf{A}_k = \begin{bmatrix} \sqrt{\beta}\mathbf{A}_{k-1} \\ \underline{x}_k^{\mathrm{H}} \end{bmatrix}. \tag{4}$$

Thus, by performing the SVD of $\mathbf{A}_k$, we obtain the matrices $\mathbf{V}_k^{\mathrm{S}}$ and $\mathbf{V}_k^{\mathrm{N}}$ as the right signal and noise singular vector matrices of $\mathbf{A}_k$. The full SVD of $\mathbf{A}_k$ is given by

$$\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{H}} = \mathbf{U}_k \begin{bmatrix} \mathbf{\Sigma}_k^{\mathrm{S}} & 0 \\ 0 & \mathbf{\Sigma}_k^{\mathrm{N}} \end{bmatrix} \begin{bmatrix} \mathbf{V}_k^{\mathrm{S}} & \mathbf{V}_k^{\mathrm{N}} \end{bmatrix}^{\mathrm{H}}. \tag{5}$$

Here, $\mathbf{U}_k$ and $\mathbf{V}_k$ denote the left and right singular vector matrix, respectively. The matrix $\mathbf{A}_k$ has n nonzero singular values. The largest r of these are the signal singular values, denoted by $\sigma_k^{\mathrm{S}}(1), \ldots, \sigma_k^{\mathrm{S}}(r)$, while the remaining singular values $\sigma_k^{\mathrm{N}}(1), \ldots, \sigma_k^{\mathrm{N}}(n-r)$ are the noise singular values. Accordingly, $\mathbf{\Sigma}_k^{\mathrm{S}}=\mathrm{diag}\left(\sigma_k^{\mathrm{S}}(1), \ldots, \sigma_k^{\mathrm{S}}(r)\right)$ and $\mathbf{\Sigma}_k^{\mathrm{N}}=\mathrm{diag}\left(\sigma_k^{\mathrm{N}}(1), \ldots, \sigma_k^{\mathrm{N}}(n-r)\right)$. The left singular vector matrix $\mathbf{U}_k$ is constantly growing in size with time k. Fortunately, we need not calculate $\mathbf{U}_k$ in our applications. For DOA and frequency tracking purposes, we only need to calculate either $\mathbf{V}_k^{\mathrm{S}}$ or $\mathbf{V}_k^{\mathrm{N}}$. The singular values are usually not needed either. However, as we show in section 4, keeping track of $\mathbf{\Sigma}_k^{\mathrm{S}}$ is beneficial for tracking the number of signals r if it is unknown and/or time varying.

3. Jacobi-type noise-averaged SVD (NASVD)

The Jacobi-type NASVD algorithm [20] is a noise subspace averaged version of the full Jacobi-type SVD updating [12]. Rather than deriving the algorithm, we briefly outline it for future reference when we present the dimension tracking technique. We suppose that $\underline{x}_k$ is an $n\times 1$ dimensional data vector (snapshot of a linear sensor array or a time series data vector) obtained at the time instant k. Supposing that the data matrix at the time instant $k-1$ is decomposed as

$$\mathbf{A}_{k-1} = \mathbf{U}_{k-1} \begin{bmatrix} \mathbf{R}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathrm{R}_{k-1}\{r+1, r+1\} \cdot \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{k-1}^{\mathrm{S}} & \mathbf{V}_{k-1}^{\mathrm{N}} \end{bmatrix}^{\mathrm{H}}, \tag{6}$$

we wish to find a similar decomposition of the updated data matrix

$$\mathbf{A}_k = \begin{bmatrix} \sqrt{\beta}\mathbf{A}_{k-1} \\ \underline{x}_k^{\mathrm{H}} \end{bmatrix}. \tag{7}$$

Thereby $\mathbf{R}_{k-1}$ is a $(r+1)\times(r+1)$ upper triangular, almost diagonal matrix. This means that the norm of its off-diagonal elements is much smaller than that of the diagonal entries. Its first r diagonal entries are good approximations of the r signal singular values. The last diagonal entry $\mathrm{R}_{k-1}\{r+1, r+1\}$ is the approximate averaged noise singular value. Notice that the same value is found in the $n-r-1$ diagonal entries of the matrix $\mathrm{R}_{k-1}\{r+1, r+1\} \cdot \mathbf{I}$ in (6). $\mathbf{V}_{k-1}^{\mathrm{S}}$ is the $n\times r$ dimensional matrix of orthonormal vectors that represent good approximations of the right signal singular vectors of $\mathbf{A}_{k-1}$. Similarly, $\mathbf{V}_{k-1}^{\mathrm{N}}$ is the approximate $n\times(n-r)$ dimensional matrix of the right noise singular vectors. The vectors of $\mathbf{V}_{k-1}^{\mathrm{N}}$ are orthonormal as well and they are also orthogonal to the signal singular vectors of $\mathbf{V}_{k-1}^{\mathrm{S}}$. We say that the columns of $\mathbf{V}_{k-1}^{\mathrm{S}}$ span the signal subspace, while the columns of $\mathbf{V}_{k-1}^{\mathrm{N}}$ span the noise subspace. Notice that the noise subspace is spherical since the noise singular values corresponding to the vectors of $\mathbf{V}_{k-1}^{\mathrm{N}}$ all equal $\mathrm{R}_{k-1}\{r+1, r+1\}$.

$\mathbf{U}_{k-1}$ is the approximate matrix of left singular vectors that we are not interested in tracking. Clearly, equation (6) is an approximate SVD of $\mathbf{A}_{k-1}$. The constant β ($0<\beta<1$) is the forgetting factor and it provides downweighting of past data samples to allow tracking nonstationarities in the signal.

In subspace tracking applications we are interested in tracking only the $n\times r$ dimensional matrix of right signal singular vectors $\mathbf{V}_k^{\mathrm{S}}$ because its columns span the signal subspace. However, having in mind the DOA or frequency retrieval applications, tracking the $(r+1)\times(r+1)$ upper triangular matrix $\mathbf{R}_k$ is also beneficial. Squares of the diagonal entries of $\mathbf{R}_k$ are good estimates of the power in the signal components. Also, $\mathrm{R}_{k-1}^2\{r+1,r+1\}$ is an estimate of the noise power in the system. Furthermore, the diagonal entries of $\mathbf{R}_k$ can be used to adaptively estimate the dimension r of the signal subspace as we will show in the next section.

The Jacobi-type NASVD algorithm updates $\mathbf{R}_k$ and $\mathbf{V}_k^{\mathrm{S}}$ in the following manner. We first compute the vector of inner products $\underline{z}_k^{\mathrm{S}}=\left(\mathbf{V}_{k-1}^{\mathrm{S}}\right)^{\mathrm{H}}\underline{x}_k$ and form the vector

$$\underline{z}_k=\begin{bmatrix}\underline{z}_k^{\mathrm{S}}\\ z_k^{\mathrm{N}}\end{bmatrix}=\begin{bmatrix}\underline{z}_k^{\mathrm{S}}\\ \sqrt{\|\underline{x}_k\|^2-\left\|\underline{z}_k^{\mathrm{S}}\right\|^2}\end{bmatrix}. \quad (8)$$

Using QR-decomposition

$$\begin{bmatrix}\sqrt{\beta}\mathbf{R}_{k-1}\\ \underline{z}_k^{\mathrm{H}}\end{bmatrix}=\mathbf{Q}_k\begin{bmatrix}\mathbf{R}_k'\\ \underline{0}^{\mathrm{T}}\end{bmatrix}, \quad (9)$$

we obtain the upper triangular matrix $\mathbf{R}_k'$. We further diagonalize $\mathbf{R}_k'$ with a series of r Jacobi rotations to get

$$\mathbf{R}_k=\mathbf{\Theta}_{(k,r)}\cdots\mathbf{\Theta}_{(k,1)}\mathbf{R}_k'\mathbf{\Phi}_{(k,1)}\cdots\mathbf{\Phi}_{(k,r)}. \quad (10)$$

The complex Jacobi rotation pair [23] $\mathbf{\Theta}_{(k,i)}$ and $\mathbf{\Phi}_{(k,i)}$ is applied on the i-th and $(i+1)$-st row and column of $\mathbf{R}_k'$ to introduce a zero in the $(i,i+1)$-st entry of $\mathbf{R}_k'$, just as in the original full Jacobi-type SVD updating algorithm [12]. However, while in the original Jacobi-type SVD strictly outer rotations are used for better convergence properties [12], [24], [25], in the NASVD algorithm only the first $r-1$ rotations are outer rotations. To see why the last rotation is an inner one, notice that due to their "mixing" properties [24], [25], [26], outer rotations cyclicly permute the singular values along the diagonal entries of $\mathbf{R}_k$. Since we want the last diagonal entry of $\mathbf{R}_k$ to be always occupied by the averaged noise singular value, we choose the r-th Jacobi rotation to be an inner one. The matrix $\mathbf{R}_k$ has a smaller off-diagonal norm than $\mathbf{R}_k'$ due to the applied Jacobi rotations. Thus $\mathbf{R}_k$ is an upper triangular, almost diagonal matrix. Of course, we can further reduce the off-diagonal norm of $\mathbf{R}_k$ by applying another sweep of Jacobi rotations to $\mathbf{R}_k$ in the same manner as described above. We will, however, use only one sweep per update in our NASVD subspace tracking algorithm.

After the Jacobi rotations have been applied, the sphericity of the noise subspace is destroyed. That is, the newly obtained noise singular value $\mathrm{R}_k\{r+1,r+1\}$ no longer equals $\sqrt{\beta}\mathrm{R}_{k-1}\{r+1,r+1\}$, which is the value of the remaining $n-r-1$ noise singular values. To force sphericity on the noise subspace, we need to reaverage the squares of the noise singular values, i.e. noise eigenvalues of a corresponding sample correlation matrix. Thus we redefine the averaged noise singular value

$$\mathrm{R}_k\{r+1,r+1\}:=\sqrt{\frac{\mathrm{R}_k^2\{r+1,r+1\}+(n-r-1)\beta\mathrm{R}_{k-1}^2\{r+1,r+1\}}{n-r}}. \quad (11)$$

To update the matrix of right signal singular vectors $\mathbf{V}_k^{\mathrm{S}}$, we first form the normalized projection of $\underline{x}_k$ on the noise subspace

$$\underline{v}_{k-1}^{\mathrm{N}}=\frac{\underline{x}_k-\mathbf{V}_{k-1}^{\mathrm{S}}\underline{z}_k^{\mathrm{S}}}{z_k^{\mathrm{N}}}. \quad (12)$$

We refer to $\underline{v}_{k-1}^{\mathrm{N}}$ as the averaged noise singular vector. It is exactly this vector that interacts with the signal subspace in the updating process [13]. As a consequence, we update the matrix $\mathbf{V}_k^{\mathrm{S}}$ as

$$\left[\begin{array}{cc}\mathbf{V}_k^{\mathrm{S}} & \tilde{\underline{v}}_k^{\mathrm{N}}\end{array}\right]=\left[\begin{array}{cc}\mathbf{V}_{k-1}^{\mathrm{S}} & \underline{v}_{k-1}^{\mathrm{N}}\end{array}\right]\mathbf{\Phi}_{(k,1)}\cdots\mathbf{\Phi}_{(k,r)}. \quad (13)$$

Notice that for subspace tracking purposes alone, we don't need to calculate the vector $\tilde{\underline{v}}_k^{\mathrm{N}}$. However, as we will show in the next section, the vector $\tilde{\underline{v}}_k^{\mathrm{N}}$

plays an important role in our subspace dimension updating scheme.

With the updated matrices $\mathbf{R}_k$ and $\mathbf{V}_k^{\mathrm{S}}$, we execute a new iteration step of the NASVD updating process as soon as the new data vector $\underline{x}_{k+1}$ becomes available. As a final remark in this section we note that the algorithm converges for any diagonal starting matrix $\mathbf{R}_0$ whose diagonal entries are non-negative, and for any matrix $\mathbf{V}_0^{\mathrm{S}}$ with orthonormal columns.

4. Subspace Dimension Tracking

Estimation of the signal subspace dimension is important for accurate subspace tracking. Most subspace tracking algorithms assume to know the subspace dimension r, and in addition, consider it to be a fixed constant. Only a small number of subspace tracking algorithms are capable of on-line estimating the subspace dimension without any a priori knowledge [14], [16], [19].

In this section, we show how to modify the Jacobi-type NASVD to estimate the subspace dimension in addition to tracking the subspace itself. Thereby, the computational complexity of the algorithm remains $O(nr)$. Moreover, the parallel structure of the algorithm is not destroyed, as will be demonstrated in an efficient systolic implementation. This makes the Jacobi-type NASVD a unique parallel subspace tracking algorithm of computational complexity $O(nr)$ that is capable of tracking the signal subspace, its dimension and all signal singular values simultaneously. We discuss two ways of estimating the subspace dimension r. First, we shortly show how information theoretic criteria can be adapted to work with the Jacobi-type NASVD. This method cannot be parallelized, but it provides a useful comparison reference for our parallel rank estimating algorithm which we present immediately afterward.

4.1. AIC and MDL dimension estimation

Information theoretic criteria like the Akaike information criterion (AIC) proposed by Akaike [27] and the minimum description length (MDL) criterion proposed independently by Rissanen [28] and Schwartz [29] are popular tools for dimension estimation. They have been applied successfully to the rank estimation using the eigenvalues of a sample correlation matrix [30]. Suppose that $l(i)$ $(i = 1, 2, \ldots, n)$ denote the real, non-increasingly ordered eigenvalues of a sample correlation matrix estimated from N data vectors $\underline{x}_k$ $(k = 1, 2, \ldots, N)$. The AIC and MDL criteria are defined by

$$\mathrm{AIC}(p) = (n-p)N \ln(\alpha(p)) + p(2n-p), \tag{14}$$

$$\mathrm{MDL}(p) = (n-p)N \ln(\alpha(p)) + p(2n-p)\frac{\ln(N)}{2} \tag{15}$$

for $p = 0, 1, \ldots, n-1$, where

$$\alpha(p) = \frac{\left(\sum_{i=p+1}^{n} l(i)\right)/(n-p)}{\left(\prod_{i=p+1}^{n} l(i)\right)^{1/(n-p)}} \tag{16}$$

denotes the ratio of the arithmetic mean to the geometric mean of the smallest $n-p$ eigenvalues. The estimate of the signal subspace dimension r is given by that value of p for which the criteria are minimized.

In order to adapt these criteria to work with the NASVD algorithm, we first note that the sample correlation matrix $\hat{\mathbf{C}}_k$ in (3) is computed by an exponentially weighted sum. Correspondingly, the number of samples N in (14) to (16) has to be replaced by the effective length of the underlying exponential window

$$N(\beta, k) = \sum_{i=1}^{k} \beta^{k-i} = \frac{1-\beta^k}{1-\beta} \approx \frac{1}{1-\beta}, \quad \text{for } k \gg 1. \tag{17}$$

More importantly, we need to slightly enhance the number of singular components to be tracked in the NASVD algorithm. To see why the change is necessary, suppose that the rank is suddenly incremented by one. Accordingly, there will be an extra signal singular value to be tracked. But since this extra singular value (the last diagonal entry of $\mathbf{R}_k$) is treated as a noise singular value and reaveraged in every update, it cannot follow the sudden change of rank. To solve this problem, we propose to track an extra noise singular value and an extra noise singular vector. This extra noise

singular value will not enter the reaveraging step (11) and will therefore be free to increase abruptly in case of rank increase. To allow for this change in the algorithm, the matrix $\mathbf{R}_k$ will have to be of size $(r+2)\times(r+2)$, and the matrix $\mathbf{V}_k^{\mathrm{S}}$ will have to be of size $n\times(r+1)$, where only those singular vectors corresponding to the r largest singular values are the actual signal subspace vectors.

With the above described changes, the NASVD algorithm is well suited for rank estimation by using the AIC and MDL criteria. After each NASVD update, squares of the first $r+1$ diagonal entries of $\mathbf{R}_k$ serve as estimates for the r signal eigenvalues and the first noise eigenvalue

$$\hat{l}_k(i) = \mathrm{R}_k^2\{i,i\}, \qquad (i = 1, 2, \ldots, r+1). \tag{18}$$

The remaining $n - r - 1$ noise eigenvalues are set equal to the square of the averaged noise singular value, which is estimated by the square of $\mathrm{R}_k\{r+2, r+2\}$.

$$\hat{l}_k(i) = \mathrm{R}_k^2\{r+2, r+2\}, \qquad (i = r+2, r+3, \ldots, n). \tag{19}$$

Using these eigenvalue estimates instead of $l(i)$ and the effective window length $N(\beta, k)$ in place of N, Eqs. (14) to (16) will return estimates of the actual rank. This procedure is similar to that proposed in [16].

4.2. Threshold comparison dimension tracking

Applying the information theoretic criteria for subspace dimension tracking serves as a bottleneck in parallel implementations on systolic arrays. This is because both AIC and MDL criteria are based on minimization of a global function whose arguments are squares of the estimated singular values of $\mathbf{A}_k$. These arguments are obtained from the diagonal entries of $\mathbf{R}_k$, which are located in non-adjacent positions in the matrix and are therefore difficult to operate on in systolic fashion. Furthermore, to find the minima of the AIC (14) or the MDL (15) criterion, we have to go through several iterations until we find that argument p for which the criteria are minimized. Clearly, this is a task that serves as a bottleneck in parallel applications. To solve this problem, we suggest a simple threshold comparison criterion.

As in the case of AIC or MDL dimension tracking, we track an extra non-averaged noise singular value and its corresponding singular vector. Thus, the matrix $\mathbf{R}_k$ is of size $(r+2)\times(r+2)$, and the matrix $\mathbf{V}_k^{\mathrm{S}}$ is of size $n\times(r+1)$. The $(r+1)$-st diagonal entry of $\mathbf{R}_k$ is the approximate non-averaged noise singular value, while the $(r+2)$-nd diagonal entry is the approximate averaged noise singular value.

Suppose that we are given a threshold T, above which we know that all signal singular values must lie. Then we formulate the threshold comparison criterion as:

IF the lowest signal singular value is less than T, decrease rank r by one;
ELSEIF the non-averaged noise singular exceeds T, increase rank r by one;
ELSE
rank r remains unchanged;
END

Now that we have the criterion, we need to specify the threshold T. One way to do this is to choose the threshold before the algorithm is started and keep it constant. Although this might actually work, it is not an interesting solution. In typical dynamic systems, the absolute noise level in the system will change (drift) with time. As the noise level in the system changes, it is logical to expect that the threshold value should change as well in order to be always greater than the expected absolute noise level. This can be easily achieved by choosing a thresholding factor $\phi > 1$, and setting the threshold

$$\mathrm{T}_k = \phi\sqrt{\lambda_k^{\mathrm{N}}}. \tag{20}$$

Here, λ_k^{N} is the noise eigenvalue defined in (1) and (2). The time index k indicates that λ_k^{N} is a function of time, i.e., that the noise level may change with time. Since λ_k^{N} is defined as the expected noise eigenvalue, in real tracking applications we would not have its value and need to estimate it from the data. This is where we can use the Jacobi-type NASVD algorithm because the square of the averaged noise singular value $\mathrm{R}_k^2\{r+2, r+2\}$ is an estimate[1] for λ_k^{N}. Thus we have

$$\mathrm{T}_k = \phi \cdot \mathrm{R}_k\{r+2, r+2\} \tag{21}$$

for our adaptively estimated threshold. Notice that we still have not specified a value for the thresholding factor ϕ. More on the optimal choice of ϕ is given in section 5. Right now we concentrate on some practical issues concerning the design of the algorithm.

In order to make the threshold comparison technique work in parallel with NASVD, we need to add a final touch to the algorithm by carefully choosing the type of Jacobi rotations we use. Our goal is to use as many outer Jacobi rotations as possible to enhance the convergence properties of the algorithm, yet we need to take care that the non-averaged and the averaged noise singular values stay locked in the $(r+1)$-st and $(r+2)$-nd diagonal entries of $\mathbf{R}_k$. We manage this by applying the Jacobi rotations in the following manner:

- Choose the first $r-1$ Jacobi rotations to be outer rotations.
- Choose the r-th Jacobi rotation to be an inner one, so that the signal singular values are kept at the first r diagonal entries of $\mathbf{R}_k$.
- Choose the $(r+1)$-st Jacobi rotation so that after the rotation is applied, $\mathrm{R}_k\{r+1, r+1\}$ is never smaller than $\mathrm{R}_k\{r+2, r+2\}$. Since $\mathrm{R}_k\{r+2, r+2\}$ will enter the reaveraging step, this prevents the larger of the two noise singular values to be reaveraged, thus making the algorithm sensitive to rank increase.

Notice that due to the "mixing" properties [24], [25], [26] of the $r-1$ outer Jacobi rotations, the r signal singular values rotate within the first r diagonal locations of $\mathbf{R}_k$. This movement is one position to the left in every update, while the singular value from the first diagonal position moves down to the r-th. Using this property, it is not necessary to search for the lowest signal singular value after every update in order to compare it to the threshold T. We simply compare only the element $\mathrm{R}_k\{r, r\}$ to the threshold because we know that at least once in every r updates the lowest signal singular value will occupy that position. This reveals that we are in a situation to perform rank deflation only once in r updates, but at least we are sure that the algorithm will certainly detect rank deflation although it might be with a delay of up to r cycles. On the other hand, locating the non-averaged noise singular value for threshold comparison purposes is trivial since, by construction of the Jacobi rotations, it is always at the entry $\mathrm{R}_k\{r+1, r+1\}$.

We summarize the extended Jacobi type NASVD algorithm together with the threshold comparison rank estimation in Table 1. Notice that in Table 1 we start the algorithm with an initial threshold T_0. The value of T_0 is not important as long as $\mathrm{T}_0 \geq 0$. Eventually, as the Jacobi-type NASVD converges, the threshold will converge to ϕ times the averaged noise singular value, and the rank will be then determined based on the adaptively estimated threshold from there on. The only impact that the initial value T_0 has, is that depending on how large T_0 is, the Jacobi-type NASVD takes more or less time to reach the steady state. Therefore, a good guess for T_0 will result in a faster correct initial rank determination than, say, a vastly overestimated T_0.

The above described dimension tracking algorithm is particularly useful for parallel implementations on systolic arrays for the following reasons. First, none of the operations involved with both subspace and rank updating, operate on non-adjacent matrix elements. Second, both rank inflation and deflation are executed by either appending a column to the right hand side of $\mathbf{V}_k^{\mathrm{S}}$ and $\mathbf{R}_k$ or deleting the rightmost column of $\mathbf{V}_k^{\mathrm{S}}$ and $\mathbf{R}_k$, respectively. Thus, the problem of deleting or inserting elements in the middle of a matrix is avoided. At last, the threshold comparison can be interlaced with the r-th and the $(r+1)$-st Jacobi rotations, which means that the dimension updating process can be started even before the subspace update has been completed.

5. Optimal thresholding factor

In this section we develop a tool to analyze the performance of the threshold comparison rank tracking criterion. Assuming that we know the true rank r, we will treat the estimated signal and noise singular values as random variables, and link them to the probabilities of a miss and false alarm. The probability of a miss P_M is defined as the probability that the rank is estimated to be one lower than the true rank r. The probability of false alarm P_F is defined as the probability of determining the rank to be one more than r.

Since calculating the probabilities P_M and P_F in real tracking applications is a rather compli-

Table 1. The NASVD algorithm with adaptive threshold rank estimation

Initialization

1. Input the initial threshold value $\mathrm{T}_0 \geq 0$, and a user supplied thresholding factor ϕ.
2. $r = 0 \quad , \quad \mathbf{R}_0 = \mathrm{T}_0 \cdot \mathbf{I}_{r+2} \quad , \quad \mathbf{V}_0^{\mathrm{S}} = \begin{bmatrix} 1 \\ \underline{0} \end{bmatrix}$.

Loop for $k = 1, \ldots, \infty$

1. Input $\underline{x}_k$.
2. Calculate projection of $\underline{x}_k$ on $\mathbf{V}_{k-1}^{\mathrm{S}}$: $\underline{z}_k^{\mathrm{S}} = \left(\mathbf{V}_{k-1}^{\mathrm{S}}\right)^{\mathrm{H}} \underline{x}_k$. $O\left(n(r+1)\right)$ flops
3. Calculate noise vector: $\underline{v}_{k-1}^{\mathrm{N}} = \dfrac{\underline{x}_k - \mathbf{V}_{k-1}^{\mathrm{S}} \underline{z}_k^{\mathrm{S}}}{\sqrt{\|\underline{x}_k\|^2 - \|\underline{z}_k^{\mathrm{S}}\|^2}}$. $O\left(n(r+1)\right) + O(n)$ flops
4. Form vector $\underline{z}_k^{\mathrm{H}} = \underline{x}_k^{\mathrm{H}} \begin{bmatrix} \mathbf{V}_{k-1}^{\mathrm{S}} & \underline{v}_{k-1}^{\mathrm{N}} \end{bmatrix} = \begin{bmatrix} \left(\underline{z}_k^{\mathrm{S}}\right)^{\mathrm{H}} & \sqrt{\|\underline{x}_k\|^2 - \|\underline{z}_k^{\mathrm{S}}\|^2} \end{bmatrix}$. No flops
5. QR-decomposition: $\begin{bmatrix} \sqrt{\beta}\mathbf{R}_{k-1} \\ \underline{z}_k^{\mathrm{H}} \end{bmatrix} = \mathbf{Q}_k \begin{bmatrix} \mathbf{R}_k' \\ \underline{0}^{\mathrm{H}} \end{bmatrix}$. $O\left((r+1)^2\right)$ flops
6. Jacobi rotations: $\mathbf{R}_k = \mathbf{\Theta}_{(k,r+1)} \cdots \mathbf{\Theta}_{(k,1)} \mathbf{R}_k' \mathbf{\Phi}_{(k,1)} \cdots \mathbf{\Phi}_{(k,r+1)}$. $O\left((r+1)^2\right)$ flops
   The first $r-1$ Jacobi rotations are outer rotations. The r-th Jacobi rotation is an inner one, while the $(r+1)$-st rotation is chosen such that $\mathrm{R}_k\{r+1, r+1\} \geq \mathrm{R}_k\{r+2, r+2\}$.
7. Signal subspace update: $\begin{bmatrix} \mathbf{V}_k^{\mathrm{S}} & \tilde{\underline{v}}_k^{\mathrm{N}} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{k-1}^{\mathrm{S}} & \underline{v}_{k-1}^{\mathrm{N}} \end{bmatrix} \mathbf{\Phi}_{(k,1)} \cdots \mathbf{\Phi}_{(k,r+1)}$. $O\left(n(r+1)\right)$ flops
8. Reaverage to maintain noise subspace sphericity:
   $\mathrm{R}_k\{r+2, r+2\} := \sqrt{\dfrac{\mathrm{R}_k^2\{r+2, r+2\} + (n-r-2)\beta \mathrm{R}_{k-1}^2\{r+2, r+2\}}{n-r-1}}$. $O(1)$ flops
9. Dimension updating by threshold comparison: $O(1)$ flops

   IF $\mathrm{R}_k\{r, r\} < \mathrm{T}_{k-1}$ perform rank deflation:
   - Set $\mathrm{R}_k\{r+1, r+1\} := \sqrt{\dfrac{\mathrm{R}_k^2\{r+1, r+1\} + (n-r-1)\mathrm{R}_k^2\{r+2, r+2\}}{n-r}}$.
   - Delete the rightmost column of $\mathbf{V}_k^{\mathrm{S}}$ and $\mathbf{R}_k$.
   - Set $r := r - 1$.

   ELSEIF $\mathrm{R}_k\{r+1, r+1\} > \mathrm{T}_{k-1}$ perform rank inflation:
   - Append an extra column to the right of $\mathbf{V}_k^{\mathrm{S}}$ and $\mathbf{R}_k$ by setting:

$$\mathbf{V}_k^{\mathrm{S}} := \begin{bmatrix} \mathbf{V}_k^{\mathrm{S}} & \tilde{\underline{v}}_k^{\mathrm{N}} \end{bmatrix} \quad \text{and} \quad \mathbf{R}_k := \begin{bmatrix} \mathbf{R}_k & \underline{0} \\ \underline{0}^{\mathrm{H}} & \mathrm{R}_k\{r+2, r+2\} \end{bmatrix}.$$

   - Set $r := r + 1$.

   ELSE rank stays the same.

10. Update the threshold value: $\mathrm{T}_k := \phi\, \mathrm{R}_k\{r+2, r+2\}$.

end TOTAL: $O\left(n(r+1)\right) + O\left((r+1)^2\right)$ flops

cated task, we shall simplify the problem. First, we assume that the signal we are tracking is stationary (or at least wide sense stationary). Second, we employ a rectangular window (instead of an exponential) of length N to determine the sample correlation matrix $\hat{\mathbf{C}}$ as

$$\hat{\mathbf{C}} = \frac{1}{N} \sum_{k=1}^{N} \left(\underline{x}_k \underline{x}_k^{\mathrm{H}}\right). \tag{22}$$

We will let $N \to \infty$ in order to study the asymptotic behavior of the eigenvalues of $\hat{\mathbf{C}}$. We assume that $\underline{x}_k$ is composed of a sum of r independent sinusoids, each of amplitude A_i, where the A_i's $(1 \leq i \leq r)$ are independent Gaussian ran-

dom variables of zero mean. We also assume that $\underline{x}_k$ is corrupted by additive, spatially and temporally uncorrelated Gaussian noise of zero mean. Our final assumption is that the correlation matrix $\mathbf{C} = \mathrm{E}(\hat{\mathbf{C}})$ has r distinct and $n-r$ identical eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_r > \lambda_{r+1} = \cdots = \lambda_n = \lambda^{\mathrm{N}}$. With these assumptions, we may apply the asymptotic theory [32] for the characteristic roots (eigenvalues) $l_1 \geq l_2 \geq \cdots \geq l_n$ of the sample correlation matrix $\hat{\mathbf{C}}$ of a jointly Gaussian distributed vector $\underline{x}_i$.

We shall use the following notation in this section. With σ_r we will denote the lowest signal singular value ($\sigma_r^2 = l_r$). σ_{NA} will stand for the non-averaged noise singular value ($\sigma_{NA}^2 = l_{r+1}$), while σ_A will denote the averaged noise singular value ($\sigma_A^2 = \dfrac{1}{n-r-1}\sum_{i=r+2}^{n} l_i$).

5.1. *Probability of a miss P_M*

Assuming that the true rank is r, the probability of a miss is defined as the probability that the estimated rank is $r-1$. In our threshold rank estimation scheme, this happens when σ_r is lower than the threshold $T = \phi \cdot \sigma_A$, i.e.,

$$P_M(\phi) = \mathrm{P}\left\{\sigma_r \leq \phi \cdot \sigma_A\right\}. \tag{23}$$

Since σ_A^2 is a good estimate[2] of λ^{N}, we have

$$P_M(\phi) \approx \mathrm{P}\{\sigma_r^2 \leq \phi^2 \lambda^{\mathrm{N}}\} = \mathrm{P}\{l_r \leq \phi^2 \lambda^{\mathrm{N}}\}. \tag{24}$$

We now apply the asymptotic result [32] that l_r has an asymptotically normal distribution with mean λ_r and variance $\sigma_\lambda^2 = \dfrac{2\lambda_r^2}{N}$. Thus, we can easily find the probability of a miss as

$$\begin{aligned} P_M(\phi) &\approx \int_{-\infty}^{\phi^2\lambda^N} \frac{1}{\sqrt{2\pi\sigma_\lambda^2}} \mathrm{e}^{-\frac{(x-\lambda_r)^2}{2\sigma_\lambda^2}} \mathrm{d}x \\ &= \int_{\lambda_r-\phi^2\lambda^N}^{\infty} \frac{1}{\sqrt{2\pi\sigma_\lambda^2}} \mathrm{e}^{-\frac{x^2}{2\sigma_\lambda^2}} \mathrm{d}x \\ &= \frac{1}{2}\,\mathrm{erfc}\left(\frac{\sqrt{N}}{2}\left[1 - \frac{\phi^2\lambda^N}{\lambda_r}\right]\right). \end{aligned} \tag{25}$$

Since

$$\frac{\lambda_r}{\lambda^N} = 1 + SNR_r, \tag{26}$$

where SNR_r is the ratio of powers of the r-th signal mode and noise, equation (25) becomes

$$P_M(\phi) \approx \frac{1}{2}\,\mathrm{erfc}\left(\frac{\sqrt{N}}{2}\left[1 - \frac{\phi^2}{1+SNR_r}\right]\right). \tag{27}$$

Thus, we see that if the SNR is smaller, in order to maintain the same P_M, we need to decrease ϕ.

5.2. *The probability of false alarm P_F*

Unfortunately, we were not able to find an expression for the probability of false alarm, but at least we can derive an upper bound for P_F. The probability of false alarm is the probability that the rank is estimated one higher than the true rank r. According to our threshold comparison criterion, this occurs when the non-averaged noise singular value σ_{NA} exceeds the threshold $\mathrm{T}_0 = \phi \cdot \sigma_A$. We shall now show that

$$\begin{aligned} P_F(\phi) &= \mathrm{P}\left\{\sigma_{NA} > \phi \cdot \sigma_A\right\} \leq \\ &\mathrm{P}\left\{\chi^2_{\frac{1}{2}(n-r+2)(n-r-1)} \geq \eta(\phi)\right\} = P_\eta(\phi), \end{aligned} \tag{28}$$

where $\chi^2_{\frac{1}{2}(n-r+2)(n-r-1)}$ is a χ^2-distributed random variable with $\frac{1}{2}(n-r+2)(n-r-1)$ degrees of freedom, and $\eta(\phi)$ is the solution of

$$\frac{\left[1 + \dfrac{\phi^2 - 1}{n-r}\right]^{n-r}}{\phi^2} = \mathrm{e}^{\eta(\phi)/N}. \tag{29}$$

Proof:: From [32] we have that $L(r)$ defined using the ratio $\alpha(r)$ (16) as

$$\begin{aligned} L(r) &= (n-r)N \ln\left(\alpha(r)\right) \\ &= -N \ln\left(\prod_{i=r+1}^{n} l_i\right) + N(n-r) \ln\left(\frac{\sum_{i=r+1}^{n} l_i}{n-r}\right) \end{aligned} \tag{30}$$

is an asymptotically χ^2-distributed random variable with $\frac{1}{2}(n-r+2)(n-r-1)$ degrees of freedom.

Thus we have

$$\begin{aligned} P_\eta(\phi) &= \mathrm{P}\{L(r) \geq \eta(\phi)\} \\ &= \mathrm{P}\left\{\chi^2_{\frac{1}{2}(n-r+2)(n-r-1)} \geq \eta(\phi)\right\}. \end{aligned} \tag{31}$$

We now need to show that $P_\eta(\phi) \geq P_F(\phi)$. Since $\sigma^2_{NA} = l_{r+1}$ and $\sigma^2_A = \dfrac{1}{n-r-1}\sum_{i=r+2}^{n} l_i$, we have that $\sigma^2_{NA} + (n-r-1)\sigma^2_A = \sum_{i=r+1}^{n} l_i$. Substituting this into (30), we find that

$$\begin{aligned} L(r) = N(n-r-1)\ln K - N\ln\left[\sigma^2_{NA}\sigma_A^{2(n-r-1)}\right] \\ + N\cdot(n-r)\ln\left[\frac{\sigma^2_{NA} + (n-r-1)\sigma^2_A}{n-r}\right], \end{aligned} \tag{32}$$

where

$$N\cdot(n-r-1)\ln K \geq 0 \tag{33}$$

since K is the ratio of the arithmetic and the geometric mean of the lowest $(n-r-1)$ eigenvalues of $\hat{\mathbf{C}}$. Consequently, we have

$$\begin{aligned} L(r) \geq -N\ln\left[\sigma^2_{NA}\sigma_A^{2(n-r-1)}\right] + \\ N\cdot(n-r)\ln\left[\frac{\sigma^2_{NA} + (n-r-1)\sigma^2_A}{n-r}\right]. \end{aligned} \tag{34}$$

Therefore,

$$P_\eta(\phi) = \mathrm{P}\{L(r) \geq \eta(\phi)\} \tag{35}$$

$$\geq \mathrm{P}\left\{ N(n-r)\ln\left[\frac{\sigma^2_{NA} + (n-r-1)\sigma^2_A}{n-r}\right] - N\ln\left[\sigma^2_{NA}\sigma_A^{2(n-r-1)}\right] > \eta(\phi)\right\} \tag{36}$$

$$= \mathrm{P}\left\{\frac{\left[1+\frac{\sigma^2_{NA}/\sigma^2_A - 1}{n-r}\right]^{n-r}}{\sigma^2_{NA}/\sigma^2_A} \geq \mathrm{e}^{\eta(\phi)/N}\right\}. \tag{37}$$

Since

$$\Gamma\left(\frac{\sigma_{NA}}{\sigma_A}\right) = \frac{\left[1+\frac{\sigma^2_{NA}/\sigma^2_A - 1}{n-r}\right]^{n-r}}{\sigma^2_{NA}/\sigma^2_A} \tag{38}$$

is a monotonically increasing function of the ratio $\dfrac{\sigma_{NA}}{\sigma_A}$, for $\dfrac{\sigma_{NA}}{\sigma_A} \geq 1$, we have that (37) is equivalent to

$$P_F(\phi) = \mathrm{P}\{\sigma_{NA} \geq \phi\cdot\sigma_A\}, \tag{39}$$

where ϕ is obtained by substituting $\phi = \dfrac{\sigma_{NA}}{\sigma_A}$ in (38) and solving $\Gamma(\phi) = \mathrm{e}^{\eta(\phi)/N}$. ■

5.3. Asymptotically optimal thresholding factor ϕ_{opt}

Now that we have a way of determining P_M and P_F (actually just an upper bound for P_F) from ϕ, we can make a plot of $1-P_M$ over P_F, for different values of ϕ. This is similar to a receiver operating characteristic, typically drawn for receivers when we are trying to detect signals based on the Neyman-Pearson strategy and a binary hypothesis, see [33] for example. We draw a curve $1-P_M$ versus P_F (actually $1-P_M$ versus P_η) for different values of r in the range $0 \leq r \leq r_{max}$, where r_{max} is the maximal anticipated signal subspace rank. Examples of these plots are shown in Figure 1.b. Notice that the plot for $r = 0$ does not make sense, since for $r = 0$, $P_M = 0$. Nevertheless, we plotted this curve according to equations (27), (28) and (29) just to establish what the dependence on the value of r is. Figure 1.a shows the plot of ϕ for different values of P_F. Thus, from figure 1.a, we find the P_F that corresponds to a given ϕ and r, and then, from figure 2.b, we find the quantity for P_M. Then we choose that value ϕ that best satisfies our criteria for how large P_F and P_M should be. For example, we may choose the optimal value ϕ_{opt} as that value ϕ, for which

$$P_M(\phi) = P_F(\phi). \tag{40}$$

The plot of the line $P_M = P_F$ is shown in figure 1.b. Intersections with the plots of $1 - P_M$ versus P_F represent optimal points for different values of r. Tracing now back to figure 1.a, we

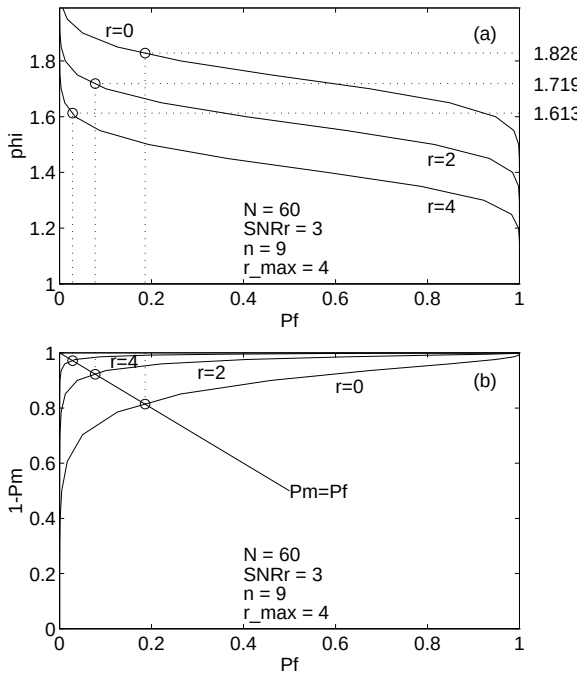

Fig. 1. (a): Thresholding factor ϕ versus the probability of false alarm P_F for different values of r. (b): $1 - P_M$ versus P_F for different values of r. Intersections with the line $P_M = P_F$ determine the optimal values of ϕ.

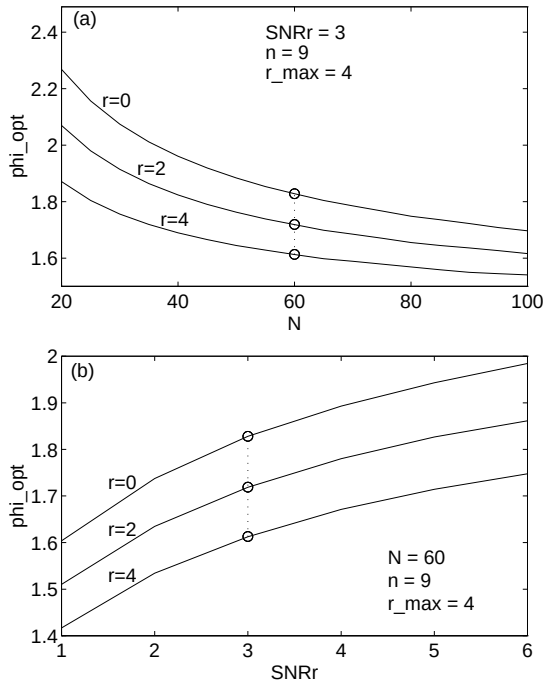

Fig. 2. (a): Asymptotically optimal thresholding factor ϕ versus the size of the rectangular window N for different values of the true rank r. (b): Asymptotically optimal thresholding factor ϕ versus the signal to noise ratio SNR_r for different values of the true rank r.

find the optimal values of ϕ. They are: $\phi = 1.828$ for $r = 0$, $\phi = 1.719$ for $r = 2$, and $\phi = 1.613$ for $r = r_{max} = 4$. Since we want to make ϕ a constant (independent of r), we choose that value ϕ to allow for optimal detection of the largest anticipated number of signal components, that is, we choose our ϕ_{opt} as $\phi_{opt}(r_{max}) = 1.613$. An alternative method would be to let ϕ depend on the currently estimated r, but that complicates slightly the criterion.

In figures 2.a and 2.b, we illustrate how the optimal value ϕ_{opt} depends on N, r, and SNR_r. In figure 2.a, we fixed SNR_r and n, while we changed N and plotted ϕ_{opt} for different values of r, ranging between 0 and $r_{max} = 4$. The plot reveals that ϕ_{opt} decreases as both N and r (or equivalently r_{max}) increase. The circled points in Figure 2 correspond to the circled points in Figure 1. Figure 2.b reveals that ϕ_{opt} increases as the signal to noise ratio SNR_r increases.

5.4. *Optimal thresholding factor for real tracking environments*

Since a real tracking environment differs from the asymptotic environment for which we derived the expressions for P_M and P_F, one may ask how useful is the asymptotic value ϕ_{opt} in real tracking applications when we made so many assumptions. As a reminder, in our asymptotic curves, we assumed:

1. Stationary signals,
2. Rectangular window of size $N \to \infty$,
3. $n - r \gg 1$,
4. $P_F(\phi) = P_\eta(\phi)$,
5. $\sigma_A^2 \approx \lambda^N$,
6. Jacobi-type SVD $\approx$ SVD.

In addition, in real tracking applications, the system designer cannot count on knowing the value for SNR_r a priori. SNR_r is the ratio of the lowest signal eigenvalue and the noise eigenvalue. Since SNR_r depends on the real SNR (defined as the power of a source over the power of noise) as well as on the problem dimension n and the

relative distance between the tracked frequencies (or DOAs) of the r signal components (which are unknown a priori), it is impossible to determine SNR_r as a function of the real SNR, n and r. Also, all the asymptotic probability distributions were derived for the batch eigenvalue decomposition case, and these are not necessarily the same for recursive updating schemes. Therefore, *quantitatively*, the asymptotic results from the previous subsection are not accurate since they do not correspond to real subspace tracking conditions. A better value for ϕ_{opt} needs to be reached by running a few tracking experiments for a fixed n, r_{max}, β and a minimum anticipated real SNR, and then choosing that ϕ_{opt} that provides the best balance between P_M and P_F.

By running these experiments, we found out that, *qualitatively*, the value for ϕ_{opt} in real tracking applications obeys laws similar to those in the asymptotic case. That is,

1. As n increases, ϕ_{opt} increases.
2. As β increases (or as the equivalent window length $N(\beta) = \dfrac{1}{1-\beta}$ increases), ϕ_{opt} decreases.
3. As the SNR increases, ϕ_{opt} increases.
4. As r increases (or if r_{max} increases to provide for the worst case), ϕ_{opt} decreases.

Hence, the asymptotic theory provides qualitative guidelines on how to choose ϕ_{opt}. Based on these observations, we fine-tune ϕ until the rank estimation performance reaches a satisfactory level. An illustration of the dependencies of ϕ_{opt} on other system parameters (specifically β and r_{max}) is given in section 7, where we present subspace and dimension tracking results.

6. Systolic Implementation

In this section we briefly demonstrate that the Jacobi-type NASVD algorithm is executable in systolic architecture. Detailed descriptions of processor elements and the timing mechanism reach beyond the scope of this paper due to limited space. We will therefore stay on the block-array level and concentrate on following the data flow and describing operations performed by each separate group of processor elements, rather than by each individual processing cell. We are using the architecture proposed by Dowling et al. [18], where their triangular QRD array is substituted by Moonen's triangular array for performing QR-decomposition interlaced with Jacobi rotations [34]. For a detailed description of the array, see [21].

In Figure 3, boxes and triangles with thick frames represent hardware arrays. Data matrices and vectors are represented by boxes framed with thin lines and an arrow showing the direction of data movement. In order to illustrate in which orientations the matrices and vectors enter and leave the hardware arrays, we use a dot • to indicate the first element of vectors and the left upper corner of matrices. The short bar — connected to the dot • gives the orientation of the first row of each data matrix. Shaded regions in the pictures represent data streams and show the paths on which the data travels during the updating process. Shift registers are represented by small truncated thin-lined boxes.

The NASVD algorithm can be implemented on a systolic array similar to the one described by Dowling et al. in [18]. The full system for subspace and rank tracking is shown in Figure 3. There are two data streams shown in the figure. The darker shaded region represents the path that the singular vector matrix $\mathbf{V}_{k-1}^{\mathrm{S}}$ covers during an update. The lighter shaded path indicates the movement of all other variables, i.e. the projection vector $\underline{z}_k = [\underline{z}_k^{\mathrm{S}}\ z_k^{\mathrm{N}}]$ and the angle parameters of the Jacobi rotations $\mathbf{\Phi}_{(k,i)}$, grouped in a vector labeled $\mathbf{\Phi}_k$ in Figure 3. The system consists of three groups of processing elements. First, there is a linear array indicated by $\underline{z}_k^{\mathrm{S}}$ in Figure 3. Its inputs are the matrix $\mathbf{V}_{k-1}^{\mathrm{S}}$ and the newly arrived data vector $\underline{x}_k$, and its output $\underline{z}_k^{\mathrm{S}}=\left(\mathbf{V}_{k-1}^{\mathrm{S}}\right)^{\mathrm{H}}\underline{x}_k$ is accumulated in the array. The processor cell directly bellow the linear array is responsible for calculating the last element of the vector $\underline{z}_k$, $z_k^{\mathrm{N}}=\sqrt{\|\underline{x}_k\|^2-\|\underline{z}_k^{\mathrm{S}}\|^2}$. Upon completion of calculating the projection vector $\underline{z}_k$, it is passed through the triangular array (labeled $\mathbf{R}_{k-1}$ in Figure 3). Meanwhile, the vectors of $\mathbf{V}_{k-1}^{\mathrm{S}}$ are passed through a system of first-in-first-out shift registers of length n and are fed back to the linear array to form the product $\mathbf{V}_{k-1}^{\mathrm{S}}\underline{z}_k^{\mathrm{S}}$. This product is fed down to the cell labeled z_k^{N} to calculate the noise vector $\underline{v}_{k-1}^{\mathrm{N}} = (\underline{x}_k - \mathbf{V}_{k-1}^{\mathrm{S}}\underline{z}_k^{\mathrm{S}})/z_k^{\mathrm{N}}$. The

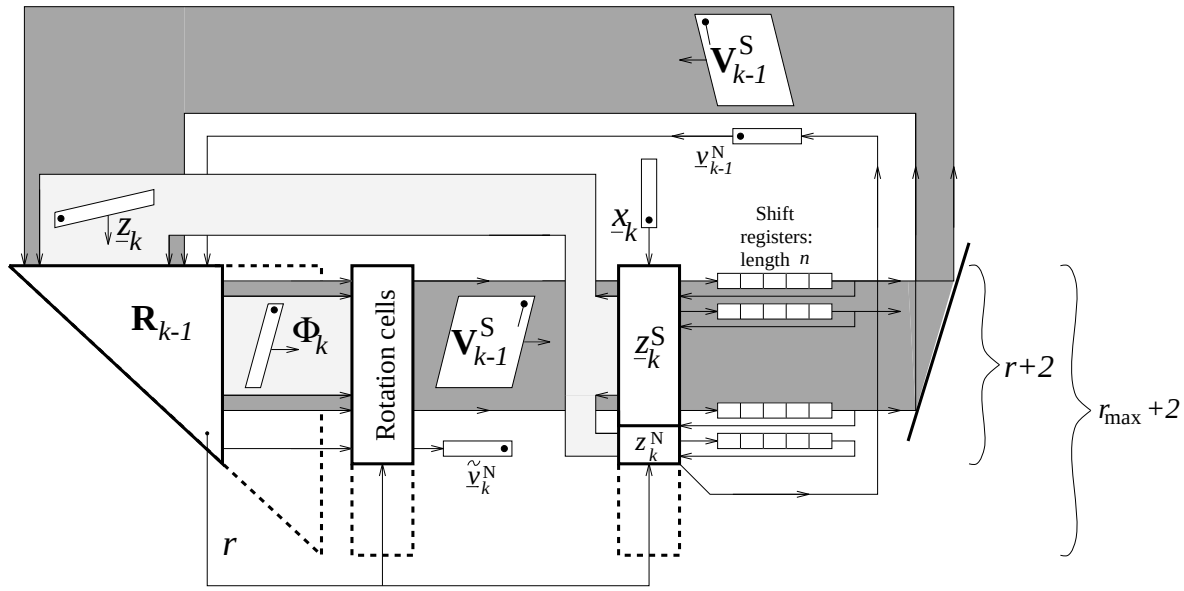

Fig. 3. Systolic NASVD system for both subspace and rank tracking.

triangular array labeled $\mathbf{R}_{k-1}$ is similar to the array proposed by Moonen et al. [12], [34], and it uses the input $\underline{z}_k$ to calculate and perform QR-decomposition and Jacobi rotations on the matrix $\mathbf{R}_{k-1}$ stored in the array. The parameters of the right Jacobi rotations $\mathbf{\Phi}_{(k,1)}, \ldots, \mathbf{\Phi}_{(k,r)}$, collectively represented by $\mathbf{\Phi}_k$ in Figure 3, are generated by the triangular array and passed rightward to the rotation cells. Jacobi rotations on the singular vectors are performed by the row of rotation cells. Before the rotations, the singular vector matrix $\mathbf{V}^{S}_{k-1}$ had to be passed through a system of shift registers (these can be built within the triangular hardware array labeled $\mathbf{R}_{k-1}$, as shown in Figure 3) to arrive at the rotation cells simultaneously with the rotations parameters $\mathbf{\Phi}_k$. The rotation cells are responsible for calculating the new singular vector matrix $\mathbf{V}^{S}_{k}$, which, together with the new data vector $\underline{x}_{k+1}$, serves as input to the linear array in the next iteration. The hardware complexity of the array is $O(r^2/2)+O(r)$ and it delivers a subspace update once in every $O(n+r)$ clock cycles. This is a bit faster than the array proposed by Dowling et al. [18] because only one row of rotation cells is needed.

We now explain how the array follows rank changes. Suppose that we have an a priori knowledge that at any point in time we are going to track at most $r_{\max}$ sources. However, if we are tracking only r sources ($r \leq r_{\max}$), not all cells need to be active. So, only the $r+2$ rows (denoted by thick lines in Figure 3) are active, while the rest (denoted by dashed lines) is idle. If the rank increases or decreases, the change is detected by threshold comparison in the last two diagonal processor cells of the triangular array, that have access to the elements $\mathrm{R}_k\{r,r\}$ and $\mathrm{R}_k\{r+1,r+1\}$. From these two cells a control signal is sent to all system elements. According to this control signal, the active portion of the systolic array is allowed to increase or decrease in the same manner as the rank itself.

Recently, Dowling et al. [19], [35] proposed a 4-level sphericalized subspace and rank tracking algorithm and a systolic array for its implementation. It is based on averaging both the signal and noise singular values, while tracking also one non-averaged signal and noise singular value for rank estimation purposes. Notice that if we are not interested in knowing all signal singular values (which is the case in some applications), we can adopt the 4-level strategy and use the Jacobi-type SVD to perform the 4×4 SVD. In that case, we can replace the triangular array with a row of processors calculating Givens rotations and a 4×4 triangular array to calculate the 4×4 SVD and track the rank by thresholding. This would lower the hardware complexity of the array to $O(r)$ while

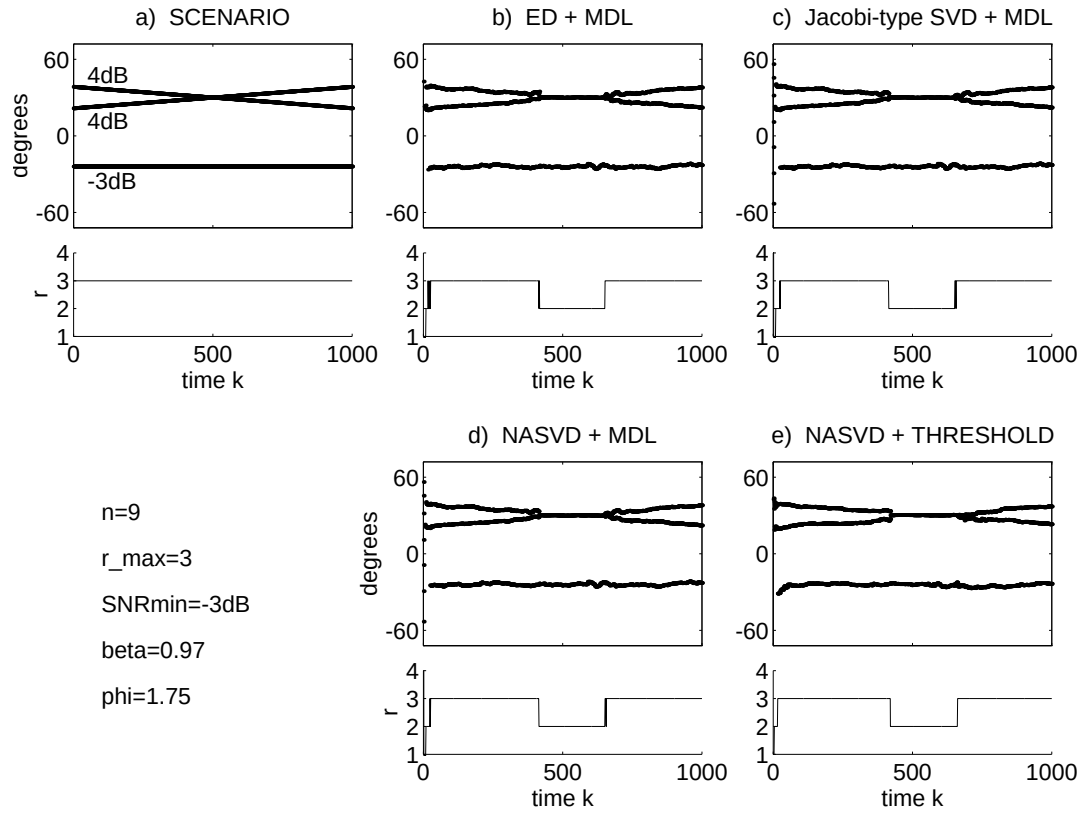

Fig. 4. Comparison of different subspace and rank tracking techniques in slowly varying signal environments with low signal-to-noise ratios. The scenario of crossing target paths tests the ability of the algorithms to resolve closely separated signal eigenvectors.

still achieving a throughput of $O(n^{-1})$. In fact, the systolic array would then be almost identical to the one proposed in [35], with the only difference being that now the Jacobi-type SVD would be used to update the 4×4 SVD and the threshold comparison criterion would be utilized for rank estimation purposes.

7. Simulation results

In this section, we use computer simulation results to compare the DOA tracking capabilities of the NASVD algorithm to other comparable subspace tracking methods. We also compare the performance of the new threshold rank tracking technique to the MDL criterion. In all simulations, a linear sensor array of $n = 9$ sensors is assumed, where the ratio of sensor distance to wave length is $\frac{1}{2}$. The noise at each sensor is Gaussian and is spatially and temporarily uncorrelated to the noise at any other sensor. The ESPRIT algorithm is used to extract the DOAs from the estimated signal subspace after every update.

7.1. Slowly varying signal environment

In Figure 4, we show the results of simulating a slowly varying signal environment. As the scenario in Figure 4.a suggests, we follow $r = 3$ targets over a time span of 1000 samples. The forgetting factor was set to $\beta = 0.97$. The DOA of the first target stays $-25°$ throughout the tracking period. The SNR of this target's signal is -3dB. We also follow two other targets, each with SNR=4dB, whose DOAs are linearly varying over the time span of 1000 snapshots. The paths of the two targets start at 25° and 40°, cross at the time instant $k = 500$ and finish at 40° and 25°, respectively. We choose the crossing path scenario to demonstrate the ability of the subspace track-

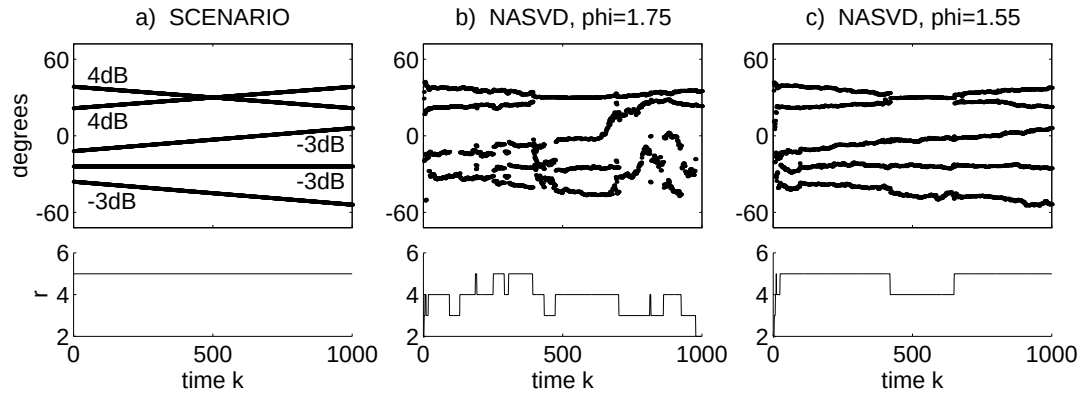

Fig. 5. Illustration of the dependence of the thresholding factor ϕ and the performance of the adaptive threshold rank estimation on the maximal anticipated number of signal components r_{max}.

ing algorithms to distinguish between closely separated signal eigenvectors.

Figure 4.b demonstrates how the batch eigenvalue decomposition combined with the MDL rank tracking criterion solves the tracking problem. Figure 4.c shows the performance of Moonen's full Jacobi-type SVD updating algorithm. In Figure 4.d, we present the results of applying the computationally efficient Jacobi-type NASVD, combined with the MDL rank tracking criterion adapted as in section IV. Finally, in Figure 4.e, the performance of the NASVD algorithm and the adaptive threshold rank estimation are presented. Clearly, we see that all four methods deliver similar results in this tracking environment. Notice that for these parameters ($n = 9$, $\text{SNR}_{min} = -3\text{dB}$, $r_{max} = 3$ and $\beta = 0.97$), the optimal thresholding factor has been determined to be $\phi = 1.75$.

To illustrate that indeed the optimal thresholding factor decreases with increasing r_{max}, we show Figure 5. There, we added two more targets to be tracked, each emitting a signal with SNR$= -3$dB, thus making $r_{max} = 5$. The previous optimal thresholding factor of $\phi = 1.75$, determined for $r_{max} = 3$, no longer delivers correct rank determination, as illustrated in Figure 5.b. However, if we lower the thresholding factor to $\phi = 1.55$ (see Figure 5.c), the rank is once again correctly estimated. This is consistent with the point made in section 5, that the optimal thresholding factor drops with increasing r_{max}.

7.2. Highly nonstationary signal environment

We simulate a highly nonstationary signal environment by letting a large number of targets suddenly appear and disappear in a time span of 1000 snapshots. We choose to put the signals from these sources in a very noisy environment (SNR≥ -2dB) to study the limits of the subspace and dimension tracking performance. Figure 6.a illustrates the scenario. To follow the rapid changes, we lowered the thresholding factor to $\beta = 0.9$ in order to enhance the influence of the recent samples on the correlation matrix estimates. Consequently, according to section 5, since β dropped, the thresholding factor for the threshold rank estimation is expected to rise. Indeed, our experiments show that the optimal thresholding factor is $\phi = 2.05$, which is larger than the optimal thresholding factors found in figures 4 and 5.

Figure 6 reveals that the MDL information criterion has problems with rank determination at these SNRs, even when applied to the batch eigenvalue decomposition (ED). Figure 6.d shows the collapse of MDL when applied to NASVD algorithm at these SNRs. This suggests that the information theoretic criteria are not optimal rank estimators, particularly in the case of noise averaged subspace trackers such as NASVD and in real (non-asymptotic), exponentially weighted tracking environments. On the other hand, the performance of the adaptive threshold rank estimation, which was carefully engineered for use with noise averaged subspace trackers in real tracking environments, shows improvements over the MDL

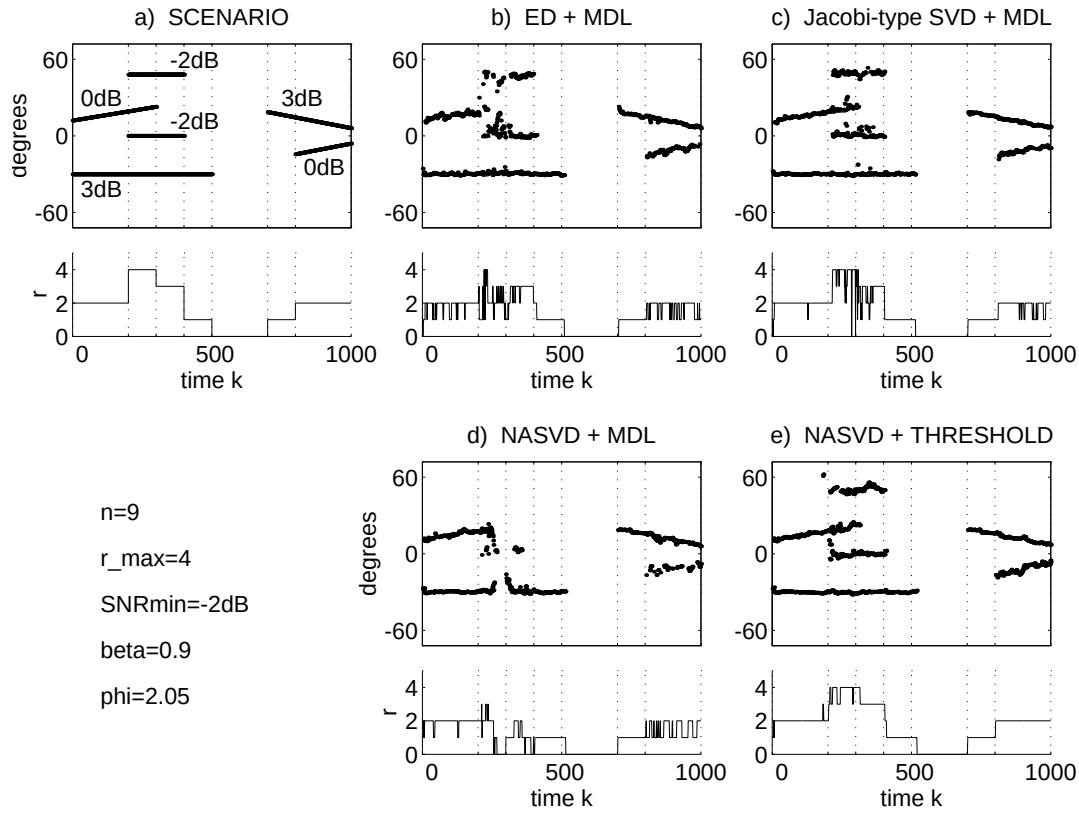

Fig. 6. Comparison of different subspace and rank tracking techniques in highly non-stationary signal environments with low signal-to-noise rations.

criterion, even when MDL is applied to batch ED, see Figure 6.e.

The last point we wish to make in this subsection is that, as often in engineering, an improvement on one side, might mean a compromise on the other. Thus, the advantage of the adaptive threshold rank estimation over information theoretic criteria, does not come without a price. The price for the adaptive threshold rank estimation is payed in spending considerable time to determine the optimal thresholding factor for a particular signal environment, either theoretically or experimentally. On the other hand, once we determine the optimal thresholding factor, we have a rank estimation method that is much simpler than information theoretic criteria. In addition, contrary to MDL or AIC, the new adaptive threshold rank tracking method works well with noise averaged subspace trackers, without forming a bottleneck in parallel systolic applications and it offers rank tracking performance that outperforms information theoretic criteria at low signal to noise ratios.

8. Conclusion

In this paper we presented a fast adaptive subspace tracking algorithm. It is based on Jacobi-type SVD updating, where the computational savings are achieved by averaging noise singular values. The method consists of interlaced QR-decomposition and Jacobi rotations. Its applications are in frequency and DOA tracking where only a few eigencomponents need to be tracked. In addition, we developed an extremely simple threshold comparison test to successfully detect the rank and thus track the signal subspace dimension. Thereby, the parallelism of the subspace tracking algorithm has not been compromised, as demonstrated in an efficient real-time systolic architecture. We presented DOA tracking simulation results to demonstrate the performance of the subspace tracking method. The re-

sults were found to be nearly identical to those obtained via batch eigenvalue decomposition and full Jacobi-type SVD. Furthermore, we tested the performance of our adaptive threshold comparison technique for signal subspace dimension tracking in comparison to information theoretic criteria (specifically MDL). It was revealed that MDL has no advantages over the much simpler threshold comparison method. In fact, adaptive threshold rank estimation performs better than MDL at very low SNRs. All of this shows why the Jacobi-type noise averaged SVD, combined with the threshold comparison subspace dimension tracking, is an attractive algorithm for real-time adaptive subspace tracking applications.

Notes

1. The average of the $n-r$ lowest eigenvalues is the maximum likelihood (ML) estimate of the noise eigenvalue λ_k^{N} [31]. Since the Jacobi-type SVD is not an exact SVD, and since $\mathrm{R}_k^2\{r+2, r+2\}$ is not obtained as an average over $n-r$ values, but rather $n-r-1$, we can only say that $\mathrm{R}_k^2\{r+2, r+2\}$ approximately equals the ML estimate of λ_k^N.
2. The maximum likelihood estimate of λ^{N} is $\hat{\lambda}_{ML}^{\mathrm{N}} = \frac{\sigma_{NA}^2 + (n-r-1)\sigma_A^2}{n-r}$. Certainly, for $n-r \gg 1$, we can say $\sigma_A^2 \approx \hat{\lambda}_{ML}^{\mathrm{N}}$.

References

1. V. F. Pisarenko, "The retrieval of harmonics from a covariance function," *Geophys. J. Roy. Astron. Soc.*, vol. 33, pp. 347–366, 1973.
2. R. O. Schmidt, "Multiple emiter location and signal parameter estimation," in *Proc. RADC Spectrum Estimation Workshop*, October 1979, pp. 243–258.
3. R. O. Schmidt, "Multiple emiter location and signal parameter estimation," *IEEE Trans. Antenn. Propagat.*, vol. AP-34, pp. 276–280, March 1986.
4. R. Kumaresan and D. W. Tufts, "Estimating the angles of arrival of multiple plane waves," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 19, pp. 134–139, 1983.
5. R. Roy and T. Kailath, "ESPRIT – estimation of signal parameters via rotational invariance techniques," *IEEE Trans. ASSP*, vol. 37, 1989.
6. M. Viberg, B. Ottersten, and T. Kailath, "Detection and estimation in sensor arrays using weighted subspace fitting," *IEEE Trans. Signal Processing*, vol. 39, pp. 2436–2449, 1991.
7. P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors," *Proc. IEEE*, pp. 1327–1343, August 1990.
8. R. J. Bunch, C. P. Nielsen, and D. C. Sorensen, "Rank-one modification of the symmetric eigenproblem," *Numer. Math.*, vol. 31, pp. 31–48, 1979.
9. N. L. Owsley, "Adaptive data orthogonalization," in *Proc. IEEE ICASSP'78*, 1978, pp. 109–112.
10. I. Karasalo, "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. ASSP*, vol. 34, pp. 8–12, February 1986.
11. J. Yang and M. Kaveh, "Adaptive eigensubspace algorithms for direction of frequency estimation tracking," *IEEE Trans. ASSP*, vol. 36, pp. 241–251, 1988.
12. M. Moonen, P. Van Dooren, and J. Vandewalle, "Updating singular value decomposition. A parallel implementation," in *Proc. SPIE Advanced Algorithms and Architectures for Signal Processing*, Vol. 1152, (San Diego), August 1989, pp. 80–91.
13. R. D. DeGroat, "Noniterative subspace tracking," *IEEE Trans. Signal Processing*, vol. 4, pp. 571–577, March 1992.
14. G. W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, pp. 1535–1541, June 1992.
15. S. Y. Kung, *Digital Neural Processing.* Englewood Cliffs: Prentice-Hall, 1993.
16. B. Yang and F. Gersemsky, "An adaptive algorithm of linear computational complexity for both rank and subspace tracking," in *Proc. IEEE ICASSP'94*, (Adelaide), April 1994, pp. IV33–IV36.
17. B. Yang, "Gradient based subspace tracking algorithms and systolic implementation," *Int. J. of High Speed Electronics and Systems*, vol. 4, pp. 203–218, 1993.
18. E. M. Dowling, L. P. Amman, and R. D. DeGroat, "A TQR-iteration based adaptive SVD for real time angle and frequency tracking," *IEEE Trans. Signal Processing*, vol. 42, pp. 914–926, April 1994.
19. E. M. Dowling, R. D. DeGroat, D. A. Linebarger, and H. Ye, "Sphericalized SVD updating for subspace tracking," in *SVD and Signal Processing: Algorithms, Applications and Architectures III*, North Holland Publishing Co., 1995.
20. A. Kavčić and B. Yang, "A new efficient subspace tracking algorithm based on singular value decomposition," in *Proceedings IEEE ICASSP'94*, (Adelaide), April 1994, pp. IV485–IV488.
21. A. Kavčić and B. Yang, "Simultaneous subspace tracking and rank estimation," in *Proc. SPIE Advanced Signal Processing Algorithms*, Vol. 2563, (San Diego), July 1995, pp. 206–217.
22. S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs: Prentice Hall, 2nd ed., 1991.
23. K. V. Fernando and S. J. Hammarling, "Systolic array computation of the SVD of complex matrices," in *Proc. SPIE Advanced Algorithms and Architectures for Signal Processing*, Vol. 696, 1986, pp. 54–61.
24. G. W. Stewart, "A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix," Tech. Report 1321, Computer Science Department, University of Maryland, 1983.
25. F. T. Luk, "A triangular processor array for computing the singular value decomposition," Tech. Report 84-625, Department of Computer Science, Cornell University, 1984.

26. J.-P. Charlier and P. Van Dooren, "A Jacobi-like algorithm for computing the generalized Schur form of a regular pencil," *J. Comp. Appl. Math.*, vol. 27, pp. 17–36, 1989.
27. A. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automatic Control*, vol. 19, pp. 716–723, 1974.
28. J. Rissanen, "Modelling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.
29. G. Schwartz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, pp. 461–464, 1978.
30. M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. ASSP*, vol. 33, pp. 387–392, 1985.
31. L. L. Scharf, *Statistical Signal Processing.* New York: Addison-Wesley Publishing Company, 1991.
32. T. W. Anderson, *An Introduction to Multivariate Statistical Analysis.* New York: John Wiley and Sons, 2nd ed., 1984.
33. H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, Vol. I. New York: John Wiley and Sons, 1968.
34. M. Moonen, *Jacobi-type Updating Algorithms for Signal Processing, Systems Identification and Control.* PhD thesis, Department of Electrical Engineering, Katholike Universiteit Leuven, Heverlee, Belgium, 1990.
35. E. M. Dowling, R. D. DeGroat, D. A. Linebarger, and Z. Fu, "Real time architecture for sphericalized SVD updating," in *SVD and Signal Processing: Algorithms, Applications and Architectures III*, North Holland Publishing Co., 1995.

Aleksandar Kavčić was born in Belgrade, Yugoslavia, in 1968. He received the Dipl.-Ing. degree in electrical engineering from the Ruhr University Bochum, Germany, in 1993. Since 1993, he has been with the Carnegie Mellon University, Pittsburgh, PA, where he is currently a Ph.D. student in the Department of Electrical and Computer Engineering. His broad research interests are in signal processing, communications and magnetic recording, while his thesis research is on modeling and detection in high density magnetic recording.

Bin Yang received the Dipl.-Ing. degree in 1986 and the Dr.-Ing. degree in 1991, both in electrical engineering from the Ruhr University Bochum, Germany.

Since 1986, he has served as a Teaching and Research staff in the Department of Electrical Engineering at the Ruhr University Bochum, Germany. His current research interests include adaptive signal processing, subspace estimation and tracking, array processing, fast and parallel algorithms, and neural networks. He is a member of IEEE.