A NEW EFFICIENT SUBSPACE TRACKING ALGORITHM BASED ON SINGULAR VALUE DECOMPOSITION

Aleksandar Kavčić [1] Bin Yang [2]

[1]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2]Department of Electrical Engineering, Ruhr University Bochum, 44780 Bochum, Germany

ABSTRACT

A new algorithm for signal subspace tracking is presented. It is based on an approximated singular value decomposition using interlaced QR-updating and Jacobi plane rotations. By forcing the noise subspace to be spherical, the computational complexity of the algorithm is brought down to $O(nr)$, where n is the problem dimension and r is the desired number of signal components. The algorithm lends itself for a very efficient systolic array implementation, resulting in a throughput of $O(n^0)$. Simulations show that the frequency tracking capabilities of the new method are at least as good as those of the computationally much more expensive exact singular value decomposition.

1. INTRODUCTION

To apply high resolution methods like MUSIC and minimum-norm to temporal and spatial domain spectral analysis, it is necessary to obtain orthonormal vectors spanning either the signal or the noise subspace. An obvious way to calculate these vectors is to perform eigendecomposition (ED) of the sample correlation matrix or singular value decomposition (SVD) of the data matrix. This approach is unacceptable for real-time tracking of nonstationary signal parameters since both ED and SVD are very time consuming.

Many different interpretations of the problem and solution approaches have been suggested in the past in order to overcome this difficulty. In all these methods, three basic trends can be observed. The first one is, of course, the reduction of computational complexity. Thus, the most computationally efficient algorithms available, i.e. [1, 2], require $O(nr)$ operations per update, where n is the problem dimension and r is the desired number of signal components. The second trend observed is to develop highly parallel subspace tracking algorithms that could be performed on systolic arrays. Most available methods can be implemented on an array with $O(n)$ processors with a throughput of $O(n^{-1})$. Moonen, Van Dooren and Vandewalle [3] and Moonen [4] have proposed an adaptive SVD method with computational complexity of $O(n^2)$ that could be performed on $O(n^2)$ processor cells having a throughput of $O(n^0)$, making the actual processing time be independent of the problem size. The third, very recent trend is to develop rank-revealing algorithms that would estimate the number of signal components in addition to tracking the signal subspace. Such procedures are the rank-revealing URV decomposition [5] and the PASTd-AIC algorithm [6] with computational complexities of $O(n^2)$ and $O(nr)$, respectively.

In our work we make use of the particularly efficient SVD-updating method proposed by Moonen, Van Dooren and Vandewalle [3]. Instead of calculating both the signal and the noise subspace, we arrange the computations so that only the signal subspace, which is usually the smaller of the two, is determined. This is accomplished by replacing the noise singular values with their average. Thus, the noise subspace is forced to be spherical, which makes it easy to deflate the problem. We name this method the noise averaged singular value decomposition (NASVD).

2. THE NASVD METHOD

Let $\mathbf{A}_k$, $\mathbf{U}_k$ and $\mathbf{V}_k$ be complex matrices of sizes $m\times n$, $m\times n$ and $n\times n$, respectively, such that

$$\mathbf{A}_k = \mathbf{U}_k \begin{bmatrix} \mathbf{R}_k & \mathbf{0} \\ \mathbf{0} & \sigma_k^{(N)}\mathbf{I}_{n-r-1} \end{bmatrix} \mathbf{V}_k^H , \qquad (1)$$

where $m\geq n$ and $\mathbf{U}_k^H\mathbf{U}_k = \mathbf{V}_k^H\mathbf{V}_k = \mathbf{I}$. The right singular vector matrix $\mathbf{V}_k$ can be written as $\mathbf{V}_k=\left[\mathbf{V}_k^{(S)}, \mathbf{V}_k^{(N)}\right]$, with $\mathbf{V}_k^{(S)}$ and $\mathbf{V}_k^{(N)}$ being the signal and the noise subspace, respectively. $\mathbf{R}_k$ is assumed to be an upper triangular, almost diagonal matrix of size $(r+1)\times(r+1)$. This means that the norm of the off-diagonal elements is much smaller than that of the diagonal entries. It is further assumed that the $(r+1)\times(r+1)$-th element of $\mathbf{R}_k$ is $\mathrm{R}_k\{r+1, r+1\}=\sigma_k^{(N)}$. Thus, equation (1) represents a good approximation of the SVD of the matrix $\mathbf{A}_k$, whose $n-r$ smallest singular values are equal to $\sigma_k^{(N)}$, the average noise singular value. The first and largest r diagonal entries of $\mathbf{R}_k$ are good approximations of the signal singular values of $\mathbf{A}_k$. Note that the left singular vector matrix $\mathbf{U}_k$ is not needed for the purpose of subspace tracking.

Our goal is to update the SVD of the matrix

$$\mathbf{A}_{k+1} = \begin{bmatrix} \sqrt{\beta}\mathbf{A}_k \\ \underline{x}_k^H \end{bmatrix} . \qquad (2)$$

Here $\underline{x}_k$ is a new data vector and β ($0<\beta<1$) is the forgetting factor. In particular, we are only interested in finding the new upper triangular, almost diagonal matrix $\mathbf{R}_{k+1}$ and the new signal subspace matrix $\mathbf{V}_{k+1}^{(S)}$. Since the noise

0-7803-1775-0/94 $3.00

subspace is spherical, one can rotate its base vectors using Householder transformations to get

$$\mathbf{A}_{k+1} \cong \begin{bmatrix} \mathbf{U}_k^* & \underline{0} \\ \underline{0}^T & 1 \end{bmatrix} \begin{bmatrix} \sqrt{\beta}\mathbf{R}_k & \mathbf{0} \\ \mathbf{0} & \sqrt{\beta}\sigma_k^{(N)}\mathbf{I}_{n-r-1} \\ \underline{z}_k^H & \underline{0}^T \end{bmatrix} \times \times \left[\mathbf{V}_k^{(S)}, \mathbf{V}_k^{(N)}\mathbf{H}_{n-r}\right]^H, \quad (3)$$

where

$$\mathbf{U}_k^* = \mathbf{U}_k \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{n-r} \end{bmatrix}. \quad (4)$$

The $(n-r)\times(n-r)$ Householder transformation $\mathbf{H}_{n-r}$ is chosen such that

$$\underline{z}_k^H \left[\mathbf{V}_k^{(S)}, \mathbf{V}_k^{(N)}\mathbf{H}_{n-r}\right] = \left[\underline{z}_k^H\mathbf{V}_k^{(S)}, \|\underline{z}_k^H\mathbf{V}_k^{(N)}\|, 0, \ldots, 0\right] = \left[\underline{z}_k^H, 0, \ldots, 0\right]. \quad (5)$$

In equation (3), an equality would hold if the off-diagonal entries of the last column of $\mathbf{R}_k$ would all be zero. Since they are only approximately equal to zero, we use the $\cong$ sign.

The next step is to use QR-decomposition to get

$$\mathbf{A}_{k+1} \cong \begin{bmatrix} \mathbf{U}_k^* & \underline{0} \\ \underline{0}^T & 1 \end{bmatrix} \mathbf{Q}_k \begin{bmatrix} \mathbf{R}_k^* & \mathbf{0} \\ \mathbf{0} & \sqrt{\beta}\sigma_k^{(N)}\mathbf{I}_{n-r-1} \\ \underline{0}^T & \underline{0}^T \end{bmatrix} \times \times \left[\mathbf{V}_k^{(S)}, \mathbf{V}_k^{(N)}\mathbf{H}_{n-r}\right]^H. \quad (6)$$

$\mathbf{R}_k^*$ is now an upper triangular matrix that can further be diagonalised using a series of p Jacobi rotations, yielding:

$$\mathbf{R}_{k+1} = \mathbf{\Theta}_{(k,p)} \cdots \mathbf{\Theta}_{(k,1)} \mathbf{R}_k^* \mathbf{\Phi}_{(k,1)} \cdots \mathbf{\Phi}_{(k,p)} \quad (7)$$

and

$$\left[\mathbf{V}_{k+1}^{(S)}, \mathbf{V}_{k+1}^{(N)}\right] = \left[\mathbf{V}_k^{(S)}, \mathbf{V}_k^{(N)}\mathbf{H}_{n-r}\right] \mathbf{\Phi}_{(k,1)} \cdots \mathbf{\Phi}_{(k,p)}. \quad (8)$$

The series of plane rotations

$$\mathbf{\Theta}_{(k,i)} = \begin{bmatrix} \mathbf{I}_{i-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} \cos\theta_{(k,i)} & \sin\theta_{(k,i)} \\ -\sin\theta_{(k,i)} & \cos\theta_{(k,i)} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{r-i} \end{bmatrix} \quad (9)$$

and

$$\mathbf{\Phi}_{(k,i)} = \begin{bmatrix} \mathbf{I}_{i-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} \cos\phi_{(k,i)} & \sin\phi_{(k,i)} \\ -\sin\phi_{(k,i)} & \cos\phi_{(k,i)} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{r-i} \end{bmatrix} \quad (10)$$

is chosen to introduce zeros at the first upper diagonal, starting at the element $\mathrm{R}_k^*\{1,2\}$ and gradually working its way down to $\mathrm{R}_k^*\{r,r+1\}$ while keeping $\mathbf{R}_k^*$ upper triangular, just as described in [3]. This procedure is to be repeated until the norm of the off-diagonal elements of $\mathbf{R}_{k+1}$ becomes much smaller than that of the diagonal entries.

Notice that only the first column of the matrix $\mathbf{V}_k^{(N)}\mathbf{H}_{n-r}$ interacts with the signal subspace $\mathbf{V}_k^{(S)}$ in equation (8). This column can be calculated as [1]

$$\underline{v}_k^{(N)} = \frac{\underline{x}_k - \mathbf{V}_k^{(S)}\underline{z}_k^{(S)}}{\sqrt{\|\underline{x}_k\|^2 - \left\|\underline{z}_k^{(S)}\right\|^2}}, \quad (11)$$

where $\underline{z}_k^{(S)} = \mathbf{V}_k^{(S)H}\underline{x}_k$. It can also be shown that $\underline{z}_k^H = \underline{x}_k^H\left[\mathbf{V}_k^{(S)}, \underline{v}_k^{(N)}\right]$. Thus, the matrix $\mathbf{V}_k^{(N)}$ is not needed at all. This enables us to reformulate equation (8):

$$\left[\mathbf{V}_{k+1}^{(S)}, \square\right] = \left[\mathbf{V}_k^{(S)}, \underline{v}_k^{(N)}\right] \mathbf{\Phi}_{(k,1)} \cdots \mathbf{\Phi}_{(k,p)}. \quad (12)$$

The sign $\square$ denotes a vector which need not be calculated.

In most practical cases, only one sweep of $p=r$ Jacobi rotations along the first upper diagonal of $\mathbf{R}_k^*$ sufficiently reduces the norm of the off-diagonal elements. All of these rotations, except the last one, should be outer rotations since they have been proven to diagonalise a matrix faster than the inner rotations. The purpose of having the last rotation be an inner one is to lock the noise singular value at the $(r+1,r+1)$-th position of the matrix $\mathbf{R}_{k+1}$, $\mathrm{R}_{k+1}\{r+1,r+1\}$. After performing this last rotation, the sphericity of the noise subspace will be destroyed because $\mathrm{R}_{k+1}\{r+1,r+1\}$ no longer equals $\sqrt{\beta}\sigma_k^{(N)}$. To force the noise subspace to be spherical again, the noise singular values are set equal to their average

$$\sigma_{k+1}^{(N)} = \mathrm{R}_{k+1}\{r+1,r+1\} := \sqrt{\frac{\mathrm{R}_{k+1}^2\{r+1,r+1\} + (n-r-1)\beta\sigma_k^{(N)2}}{n-r}}. \quad (13)$$

This formula is derived from an eigenvalue reaveraging formula often used in eigenvalue decomposition of correlation matrices. Note that singular values of a data matrix $\mathbf{A}_k$ are square roots of eigenvalues of an estimated correlation matrix $\mathbf{C}_k = \mathbf{A}_k^H\mathbf{A}_k$.

The computational complexity of the above described NASVD algorithm is $O(nr)$. Notice that the only values needed to be stored at each time step k are the right singular vector matrix $\mathbf{V}_k^{(S)}$ of size $n\times r$ and the upper triangular matrix $\mathbf{R}_k$ of size $(r+1)\times(r+1)$.

3. PARALLEL IMPLEMENTATION USING SYSTOLIC ARRAYS

The systolic array proposed for the implementation of the NASVD algorithm consists of a rectangular $n\times r$ part and a triangular $(r+1)\times(r+1)$ part containing the elements of matrices $\mathbf{V}_k$ and $\mathbf{R}_k$, respectively (see figure 1). Both, the rotations associated with the QR decomposition and the Jacobi rotations are generated at the diagonal of the triangular part. Row transformation parameters are passed on to the right, while column transformations are passed on upward. The processor cells in the upper rectangular part can only execute column rotations and propagate the rotation parameters and intermediate results. All processor

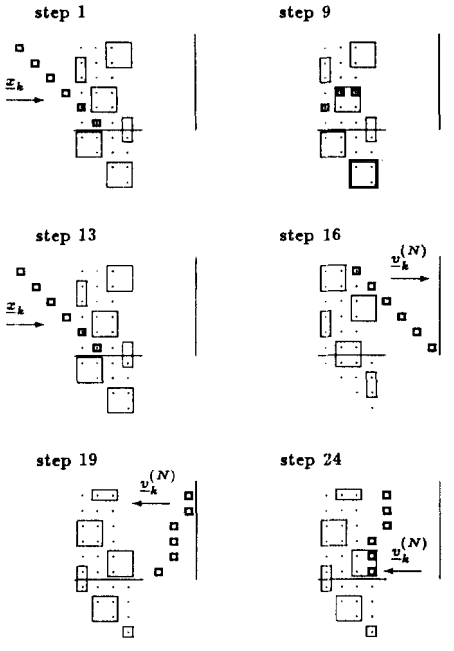

Figure 1. Data flow in the NASVD systolic array

cells are divided into four groups, corresponding to their position within the array. These four groups are cyclically activated with a period of four clock pulses. In figure 1, the active processors at each step are denoted by squares.

The structure of the NASVD systolic array is very similar to the array proposed by Moonen, Van Dooren and Vandewalle in [3] and by Moonen in [4]. Despite this similarity, the implementation of the NASVD method on the proposed array is not straightforward. The major difficulties arise in computing the noise vector $\underline{v}_k^{(N)}$. To compute this vector, the process needs to wait for some critical rotations to be calculated by the triangular part first. On the other hand, the triangular part destroys the vector $\underline{x}_k^{(S)}$, which is needed for determining the same vector $\underline{v}_k^{(N)}$ in equation (11). To overcome this difficulty, we choose to estimate the noise vector from a slightly different vector $\underline{x}_{k\|}^{(S)}$ that is available before it enters the triangular array. It turns out that when the signals parameters are slowly varying functions of time, this approach yields satisfactory results in subspace tracking applications. For a detailed description, see [4] and [7].

Without going much into detail, we show the signal flow in figure 1. Notice that the same input vector $\underline{x}_k$ enters the array twice. The first time (step 1) it is used to generate intermediate results, which are then reflected from the top and passed downward (step 9). The signal vector enters the array for the second time in step 13. This time it is used to estimate the vector $\underline{v}_k^{(N)}$, which exits the array on the right hand side in step 16. This vector is reflected from the rightmost wall in step 19. In the meantime, the triangular part is done with calculation of the Jacobi rotations, so that when the vector $\underline{v}_k^{(N)}$ enters the array from the right (step 24), the rotation $\Phi_{(k,p)}$ in equation (12) can be operated on the vector $\underline{v}_k^{(N)}$. Once again, we point out that a new input vector $\underline{x}_k$ enters the array at every fourth step, resulting in a throughput of $O(n^0)$.

4. SIMULATIONS

We show some simulation results to demonstrate the applicability and the performance of our algorithm. We use it for tracking linearly time-varying frequencies of three (r=3) complex sinusoids in additive white noise. The frequencies are normalized to lie between $-.5$ and .5. Two frequencies start at .2 and .3, cross at .25 and finish at .3 and .2 over a span of 1000 updates. The third sinusoid has a constant frequency of $-.2$. The SNR is 5 dB. The forgetting factor β is set equal to .97. Root-MUSIC is used to extract frequency information from the updated subspace.

In figure 2, plots (a) and (b) show the performance of the method proposed by Moonen, Van Dooren and Vandewalle and our shorter method. It can be seen that our NASVD method successfully tracks the signal subspace. The differences between plots (a) and (b) are due to the fact that the NASVD algorithm performed only r Jacobi rotations and not n as the longer method. Since every Jacobi rotation reduces the norm of the off-diagonal elements, the slight inaccuracy of our method is explained by the smaller number of Jacobi rotations. Plot (c) shows how the performance of our algorithm changes when 2 sweeps along the first upper diagonal of the matrix $\mathbf{R}_k^*$ are made per update. A comparison with the experiment (d), in which the exact SVD of the updated data matrix $\mathbf{A}_{k+1}$ is computed, reveals that our NASVD algorithm works at least as well as the exact SVD approach. Plot (e) shows the results obtained by simulating the work of the proposed systolic array. It can be seen that the NASVD array is able to track all three frequences. The degradation of performance in plot (e) is due to the estimation of the vector $\underline{v}_k^{(N)}$, as described in section 3.

5. SUMMARY

An efficient subspace tracking algorithm based on SVD and supposed sphericity of the noise subspace has been presented. We conclude this paper by listing the major properties of the derived NASVD method:

- The computational complexity is $O(nr)$. We note that most of the currently available subspace tracking algorithms require $O(n^2r)$, $O(n^2)$ or $O(nr^2)$ operations per update.
- With a few adaptions, the algorithm could be performed on a systolic array, which is similar in structure to the array proposed in [4] and has $O(nr)$ cells and a throughput of $O(n^0)$.
- The algorithm is able to track all signal singular values. These singular values and the average noise singular value give useful information about the power of the signal eigencomponents as well as the noise power.

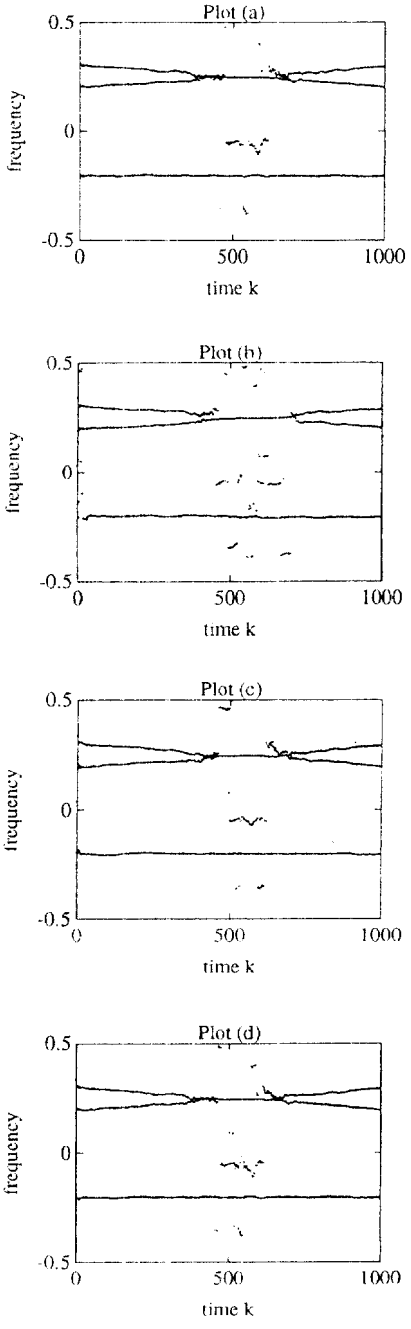

Figure 2. Frequency tracking performance
(a) Using the algorithm proposed by Moonen, Van Dooren and Vandewalle
(b) Using the NASVD algorithm
(c) Using the NASVD algorithm with 2 sweeps per update
(d) Using the exact SVD
(e) Using the proposed NASVD systolic array

Further, the calculated singular values could be used to adaptively estimate the signal subspace size r [6].

- Simulations show that the frequency tracking capabilities of the proposed method are at least as good as those of the computationally more expensive exact SVD.

REFERENCES

[1] R. D. DeGroat, "Noniterative subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, pp. 571–577, 1992.

[2] B. Yang, "Subspace tracking based on the projection approach and the recursive least squares method," in *Proc. IEEE ICASSP*, Minneapolis, Apr. 1993.

[3] M. Moonen, P. van Dooren, and J. Vandewalle, "Updating singular value decompositions. A parallel implementation.," in *Proc. SPIE Advanced Algorithms and Architectures for Signal Processing*, vol. 1152, pp. 80–91, San Diego, Aug. 1989.

[4] M. Moonen, *Jacobi-type updating algorithms for signal processing, systems identification and control*, Dissertation, Katholike Universiteit Lueven, Department of Electrical Engineering, Heverlee, Belgium, 1990.

[5] G. W. Stewart, "An updating algorithm for subspace tracking," in *IEEE Trans. Signal Processing*, vol. 40, pp. 1535–1541, 1992.

[6] B. Yang and F. Gersemsky "An adaptive algotithm of linear computational complexity for both rank and subspace tracking," to appear in *Proc. IEEE ICASSP*, Adelaide, Apr. 1994.

[7] A. Kavčić, *Untersuchung ausgewälter Verfahren zur rekursiven Eigenwertzerlegung*, Master thesis, Ruhr University Bochum, Bochum, Germany, 1993.