SVD-Updating via Constrained Perturbations with Application to Subspace Tracking

Benoît Champagne

INRS-Télécommunications, Université du Québec
Iles des Soeurs, Québec, Canada, H3E 1H6
champagne@inrs-telecom.uquebec.ca

Abstract

In this paper, we propose new algorithms for approximate updating of the singular value decomposition (SVD) of an exponentially weighted data matrix after appending a new row. The algorithms are obtained in two steps: noise subspace sphericalization is first used to deflate the problem; the right singular vectors and the singular values are then efficiently updated by means of a recently proposed constrained perturbation approach. The latter is based on Givens rotations and thus preserves the orthonormality of the updated singular vectors. The new algorithms have complexity ranging from $O(Nr)$ to $O(Nr^2)$, where N and r respectively denote the data vector and signal-subspace dimensions. Their convergence behavior in subspace tracking applications is investigated by means of the ODE method and the results are supported by computer experiments.

1. Introduction

Subspace-based signal analysis methods have proven to be of great utility in a wide variety of detection and parameter estimation problems. The distinguishing feature of these methods is the use of the eigenvalue decomposition (EVD) of the data covariance matrix to extract the desired information about the signal and noise. In practice, the EVD estimation is often realized by computing the singular values (sv) and right singular vectors (rsv) of a data matrix, whose rows are made up of successive data vectors.

In the application of subspace methods to non-stationary data, it is highly desirable to udpate the sv and rsv of the data matrix as new rows are added. Several numerical techniques do exist for the computation of the sv and rsv of an arbitrary matrix. However, direct application of these techniques from scratch to the updated data matrix at each iteration is usually not practical due to the excessive computational load involved. In recent years, new algorithms have thus been proposed for carrying out this update more efficiently based on approximations of various kinds (see for example [1-3]). Referred to as subspace trackers, these algorithms differ in their structure and complexity, the underlying principles on which they are based and the type of SVD information they compute.

In this paper, we present new algorithms for approximate updating of the sv and rsv of an exponentially weighted data matrix after appending a new row. Based on a new kind of constrained perturbation approach recently proposed in [4], these algorithms make use of sequences of Givens rotations to update the rsv so that the latter remain orthonormal at all time. The new algorithms have complexity ranging from $O(Nr)$ to $O(Nr^2)$, where N and r respectively denote the data vector and signal-subspace dimensions. Their convergence behavior in subspace tracking applications is investigated by means of the ODE method and the results are supported by computer experiments.

2. The new SVD-updating algorithms

2.1. Problem definition

Let $\mathbf{x}_k \in \mathbb{C}^N$, where $k \in \{1, 2, ...\}$ is the discrete-time index, represent a random sequence of complex data vectors upon which some processing needs to be performed. Let X_k denote the exponentially weighted data matrix defined by the data vectors via the recursion

$$X_k = \begin{bmatrix} \alpha X_{k-1} \\ \beta \mathbf{x}_k^H \end{bmatrix} \tag{1}$$

where $0 < \alpha < 1$ controls the memory of the exponential weighting and $0 < \beta < 1$ is a normalization constant such that $\alpha^2 + \beta^2 = 1$. Initialization of (1) requires the specification of X_0. This can be obtained from a previous sequence of observations or from *a priori* knowledge. To simplify the presentation, we assume that the size of X_0 is $N \times N$.

Let the singular value decomposition (SVD) of the data matrix X_k be expressed in the form

$$X_k = W_k \Sigma_k U_k^H. \tag{2}$$

In (2), W_k is a matrix of size $(N+k) \times N$ whose columns, called the left singular vectors, are orthonormal, i.e. $W_k^H W_k = I$, where I denotes an identity matrix of appropriate size; U_k is a square matrix of size N whose columns, called the right singular vectors (rsv), are also orthonormal, i.e. $U_k^H U_k = I$, so that U_k is actually a unitary matrix; and Σ_k is an $N \times N$ diagonal matrix whose entries, called the singular values (sv), are real and non-negative. We shall denote the ith column of U_k by $\mathbf{u}_{i,k}$, and the ith diagonal entry of Σ_k by $\sigma_{i,k}$. Without loss of generality, we shall assume that $\sigma_{i,k} \geq \sigma_{i+1,k}$.

The SVD (2) plays a fundamental role in the application of subspace-based signal analysis methods. Indeed, the latter requires the estimation of all (or a part) of the eigenvalue decomposition (EVD) of the data covariance matrix $R_k^o = E[\mathbf{x}_k \mathbf{x}_k^H]$. Such knowledge is directly provided by the sv and rsv of the data matrix X_k. To see this, simply note from (1) that the matrix $R_k = X_k^H X_k$ is the well-known exponentially weighted sample covariance matrix, while from (2), its EVD is given by $U_k \Sigma_k^2 U_k^H$.

In many applications involving signals in noise, the sv configuration is such that $\sigma_{1,k} \geq ... \geq \sigma_{r,k} > \sigma_{r+1,k} \approx ... \approx \sigma_{N,k}$, for some positive integer $r < N$. The rsv corresponding to the r largest (dominant) sv are then said to span the signal subspace, while the rsv corresponding to the smallest (subdominant) sv are said to span the noise subspace. The two sets of rsv are represented here by

$$U_{s,k} = [\mathbf{u}_{1,k}, ..., \mathbf{u}_{r,k}], \quad U_{n,k} = [\mathbf{u}_{r+1,k}, ..., \mathbf{u}_{N,k}]. \tag{3}$$

In this case, one is often interested in computing only the r largest (or dominant) sv and corresponding rsv.

Our aim here is to develop computationally efficient algorithms for recursively updating the r dominant sv and rsv of the data matrix X_k as a new row is added. We point out that, since the SVD information is ultimately used to estimate the corresponding EVD information of the covariance matrix R_k^o, approximation errors in the SVD computation are acceptable, provided they are masked by the inherent statistical fluctuations in the data.

2.2. Noise subspace sphericalization

The derivation of the new SVD updating algorithms begins with the application of the so-called noise subspace sphericalization,whose underlying principles can be exposed as follows [5]. Assume that at time $k-1$, the $N-r$ smallest (subdominant) sv are identical, i.e.

$$\sigma_{r+1,k-1} = ... = \sigma_{N,k-1}, \tag{4}$$

so that any orthonormal basis of the noise subspace can be used to form the SVD of X_{k-1}. If, in particular, we rotate the basis vectors $U_{n,k-1}$ so that $U_{n,k-1}^H \mathbf{x}_k$ has only one non-zero component, then the dimension of the SVD updating problem can be reduced from N to $r+1$. This technique is of considerable practical interest since in many applications, one has $r \ll N$.

Starting from (1) and expressing X_{k-1} in terms of its SVD, i.e. $X_{k-1} = W_{k-1}\Sigma_{k-1}U_{k-1}^H$, we obtain:

$$\begin{aligned} X_k &= W_a \begin{bmatrix} \alpha\Sigma_{k-1} \\ \beta\mathbf{y}_a^H \end{bmatrix} U_{k-1}^H, \quad \mathbf{y}_a = U_{k-1}^H \mathbf{x}_k & (5) \\ &= W_b \begin{bmatrix} \alpha\Sigma_{k-1} \\ \beta\mathbf{y}_b^T \end{bmatrix} D^H U_{k-1}^H, \quad \mathbf{y}_b = D^H \mathbf{y}_a & (6) \end{aligned}$$

where W_a, W_b, etc., denote matrices of appropriate sizes with orthonormal columns and D is a diagonal matrix with entries $y_{a,i}/|y_{a,i}|$, so that the components of $\mathbf{y}_b$ are real.

From (4), it follows that for any matrix $H = \text{diag}(I_r, H_n)$, where H_n is a real orthogonal matrix of dimension $(N-r)$, (6) can be expressed in the form

$$X_k = W_c \begin{bmatrix} \alpha\Sigma_{k-1} \\ \beta\mathbf{y}_c^T \end{bmatrix} H^T D^H U_{k-1}^H, \quad \mathbf{y}_c = H^T \mathbf{y}_b. \tag{7}$$

If we choose for H_n an Householder matrix such that

$$H_n^T \mathbf{y}_{b,n} = \|\mathbf{y}_{b,n}\|_2 [1, 0, ..., 0]^T, \tag{8}$$

where $\mathbf{y}_{b,n} = [y_{b,r+1}, ..., y_{b,N}]^T$, then we obtain

$$X_k = W_c \begin{bmatrix} \alpha S & 0 \\ 0 & \alpha\sigma_n I \\ \beta\mathbf{y}^T & 0 \end{bmatrix} [U_s', \mathbf{u}_{r+1}', ..., \mathbf{u}_N']^H \tag{9}$$

where $S = \text{diag}(\sigma_{1,k-1}, ..., \sigma_{r+1,k-1})$, $\sigma_n = \sigma_{r+1,k-1}$ and

$$U_s' = U_{s,k-1} D_s, \quad \mathbf{y}_s = U_s'^H \mathbf{x}_k, \tag{10}$$

$$\mathbf{x}_n = \mathbf{x}_k - U_s' \mathbf{y}_s, \quad y_{r+1} = \|\mathbf{x}_n\|, \tag{11}$$

$$\mathbf{y}^T = [\mathbf{y}_s^T, y_{r+1}], \quad \mathbf{u}_{r+1}' = \mathbf{x}_n / y_{r+1}. \tag{12}$$

Now, define $S' = \text{diag}(\sigma_1', ..., \sigma_{r+1}')$ and $V = [\mathbf{v}_1, ..., \mathbf{v}_{r+1}]$, where σ_i' and $\mathbf{v}_i$ denote the sv (in non-increasing order) and corresponding rsv of the matrix

$$Y = \begin{bmatrix} \alpha S \\ \beta\mathbf{y}^T \end{bmatrix}. \tag{13}$$

Then, from (9) and (13), we obtain

$$X_k = W_d \begin{bmatrix} S' & 0 \\ 0 & \alpha\sigma_n I \end{bmatrix} \begin{bmatrix} V^H & 0 \\ 0 & I \end{bmatrix} [U_s', \mathbf{u}_{r+1}', ..., \mathbf{u}_N']^H \tag{14}$$

which is the desired SVD expression.

In summary, the original N-dimensional problem consisting in finding the sv and rsv of X_k (1) has been deflated to one of dimension $r+1$, namely: finding the sv and rsv of Y (13). From (14), we note that $[U_{s,k}, \mathbf{u}_{r+1,k}] = [U_s', \mathbf{u}_{r+1}']V$, so that it is not necessary to compute $\mathbf{u}_{r+2}', ..., \mathbf{u}_N'$.

2.3. Constrained perturbation

To determine the sv and rsv of the deflated matrix Y (13), we use a new kind of constrained perturbation approach that has been recently proposed in [4] for the solution of the rank-one EVD modification problem. To this end, we first recast the deflated SVD update problem into an equivalent rank-one EVD update. From (13) and the definitions of V and S', we immediately obtain

$$VS'^2V^T = Y^TY = \alpha^2S^2 + \beta^2\mathbf{y}\mathbf{y}^T \tag{15}$$

The determination of the sv and rsv of Y is thus equivalent to finding the eigenvalues and eigenvectors of a diagonal matrix modified by a rank-one perturbation.

We note that in most SVD-tracking applications, the parameter $\beta^2 \ll 1$, so that the rank-one perturbation term in (15) is relatively small. In [6], this motivated the use of a first-order perturbation analysis to derive approximate solutions to (15); the validity of this approach for subspace tracking was further demonstrated by means of computer experiments and a simplified convergence study. The main limitation of a conventional perturbation analysis is its inability to produce perfectly orthonormal eigenvectors. In SVD-tracking applications, this means that even though the subspace information may be very accurate, the orthonormality error of the rsv will saturate at some non-zero level after a large number of iterations, which is questionable. In [4], an improved constrained perturbation approach was proposed to overcome this limitation. One of its main features is the use of sequences of Givens rotations to update the eigenvectors, so that orthonormality is preserved exactly. In the present SVD application, its use translates into perfectly orthonormal rsv after each update.

The main steps in the application of constrained perturbation analysis to (15) can be summarized as follows. The diagonal matrix S'^2 and the orthogonal matrix V are first expressed in terms of small, unconstrained parameters. In the case of S'^2, we write:

$$S'^2 = S^2 + \Delta, \quad \Delta = \text{diag}(\delta_1, ..., \delta_{r+1}). \tag{16}$$

Clearly, it is the parametrization of V that requires more thoughts. To this end, we first note that for V to be orthogonal, we need $\det V = \pm 1$. Without loss of generality, we assume that $\det V = 1$. With this additional restriction, V now belongs to the special orthogonal group $SO(r+1)$, also known as proper rotations. The desired parametrization is obtained by noting that any member of $SO(r+1)$ can be expressed in the form

$$V = e^{\Theta} = \sum_{k=0}^{\infty} \Theta^k/k! \tag{17}$$

where $\Theta = [\theta_{ij}]$ is a real skew-symmetric matrix of dimension $r+1$ (i.e., $\Theta^T = -\Theta$).

Using the above parametrizations of S'^2 and V, a conventional first-order perturbation analysis of the EVD problem (15) is then carried out in terms of the unconstrained parameters δ_i and θ_{ij}. Thus, substituting (16)-(17) in (15) and performing the necessary manipulations, the following expressions are obtained:

$$\delta_i = \beta^2(y_i^2 - \sigma_{i,k-1}^2) \tag{18}$$

$$\theta_{ij} = \beta^2 y_i y_j / (\sigma_{j,k-1}^2 - \sigma_{i,k-1}^2), \quad i < j \tag{19}$$

Note that since V is related to the θ_{ij} through (17), it remains orthogonal throughout the approximation process leading to (18)-(19). Although $\sigma_{1,k-1} > ... > \sigma_{r+1,k-1}$ is implicitly assumed in (19), this is not strictly necessary; repeated sv can be dealt with easily with deflation.

In [4], various approximations in terms of Givens rotations are proposed for the efficient computation of the exponential matrix V (16), when the entries of Θ have the form (19). A first approximation is given by

$$V_1 = \prod_{j>i} G_{ij}(\theta_{ij}), \tag{20}$$

where $G_{ij}(\theta)$ denotes a Givens rotation in the (i,j)-coordinate plane with rotation angle θ. Thus, for β sufficiently small, V can be expressed as the product of $r(r+1)/2$ Givens rotations with rotation angles θ_{ij} (19).

A second approximation is obtained by further assuming that the sv are well separated, i.e. $\sigma_{1,k-1}^2 \gg ... \gg \sigma_{r+1,k-1}^2$. Major simplifications then result, leading to an expression involving only $2r-1$ Givens rotations:

$$V_2 = B_r ... B_2 B_1 A_2^T ... A_r^T \tag{21}$$

$$A_i = G_{i,i+1}(\theta_{A,i}), \quad B_i = G_{i,i+1}(\theta_{B,i}) \tag{22}$$

where the rotation angles $\theta_{A,i}$ and $\theta_{B,i}$ can be computed explicitly (see also Table 3).

2.4. Description of the new algorithms

The new SVD-updating algorithms are obtained by combining the results of Sections 2.2 and 2.3. The main algorithm structure is presented in Table 1; it can be used with either one of the rsv update procedures in Table 2 or 3.

Referring to Table 1, step 1 corresponds to initialization. In the absence of *a priori* knowledge, one can select $U_{s,0}$ and $\{\sigma_{i,0}\}_{i=1}^{r+1}$ arbitrarily, subject to the constraints $U_{s,0}^H U_{s,0} = I$ and $\sigma_{i,0} > \sigma_{i+1,0}$. It is also possible to initialize the algorithms by computing (only once) the exact SVD of an initial data matrix X_0 obtained from past observations.

Step 2 refers to the main loop over the time index k. The noise subspace sphericalization is implemented in step 2-a. Updating of the rsv matrix U based on constrained perturbation is performed in step 2-b. In essence, this amounts to the transformation $U \leftarrow UV$, where V contains the rsv of the deflated matrix Y (13). We recall that two different approximations of V were given in Section 2.3, namely V_1 (20) and V_2 (21). Accordingly, two different procedures for updating U are proposed: the first one, based on V_1, is presented in Table 2 and corresponds to our first SVD-updating algorithm (SVD1); the second one, based on V_2, is presented in Table 3 and corresponds to our second algorithm (SVD2). Updating of the sv σ_i is performed in step 2-c. Note that after the sv have been updated according to (16) and (18), the $N-r$ subdominant sv are no longer equal (i.e. $\sigma_{r+1} > \sigma_n$) and must be re-averaged in preparation for the next iteration. Finally, in step 2-d, the sv are compared and, if necessary, rearranged in decreasing order; a corresponding permutation is applied to the rsv.

Algorithms SVD1 and SVD2 can be used to track the r dominant sv and rsv and the average value of the subdominant sv of the data matrix X_k (1). These algorithms maintain a true SVD structure at all time: the sv $\sigma_{i,k}$ remain non-negative and the orthonormality of the rsv $\mathbf{u}_{i,k}$ is preserved due to the use of Givens rotations.We point out that even though the derivation of V_2 is based on the assumption $\sigma_i^2 \gg \sigma_{i+1}^2$, the corresponding algorithm SVD2 is robust and generally converges even with closely spaced sv.

For complex data, SVD1 requires $0.75Nr^2 + O(Nr)$ complex operations (cops) per iteration, where one cop is defined as four real multiplications and four real additions; SVD2 requires only $5.75Nr + O(N)$ cops per iteration but its implementation is slightly more elaborate. In terms of convergence performance, the algorithms have comparable behaviors, although SVD1 is slightly more robust in certain difficult situations.

Step	Operation
1	Initialization:
	$\alpha \leftarrow$ exponential window parameter
	$\beta \leftarrow \sqrt{1-\alpha^2}$
	$U_s \leftarrow U_{s,0}, \quad N \times r$
	$\sigma_i \leftarrow \sigma_{i,0}, \quad i = 1, ..., r+1$
2	Main loop over discrete-time index k:
	for $k = 1, 2, ...$
(a)	Noise subspace sphericalization:
	$\mathbf{y}_s \leftarrow U_s^H \mathbf{x}_k$
	$D_s \leftarrow \mathrm{diag}(y_i/\lvert y_i\rvert; i = 1, ..., r)$
	$\mathbf{y}_s \leftarrow D_s^H \mathbf{y}_s$
	$U_s \leftarrow U_s D_s$
	$\mathbf{x}_n \leftarrow \mathbf{x} - U_s \mathbf{y}_s$
	$y_{r+1} \leftarrow \lVert\mathbf{x}_n\rVert_2$
	$\mathbf{u}_{r+1} \leftarrow \mathbf{x}_n / y_{r+1}$
	$U \leftarrow [U_s, \mathbf{u}_{r+1}]$
	$\mathbf{y}^T \leftarrow \beta[\mathbf{y}_s^T, y_{r+1}]$
(b)	Update rsv via constrained perturbation:
	select desired algorithm from Table 2 or 3
(c)	Update sv via constrained perturbation:
	$\sigma_n \leftarrow \alpha\sigma_{r+1}$
	$\sigma_i \leftarrow \alpha\sigma_i\sqrt{1+(y_i/\alpha\sigma_i)^2}, \quad i = 1, ..., r+1$
	$\sigma_{r+1} \leftarrow [\sigma_{r+1} + (N-r-1)\sigma_n]/(N-r)$
(d)	Rearrange sv (and rsv) in decreasing order
	end

Table 1. New SVD-updating algorithms for tracking r dominant sv and rsv (use with either one of rsv updates in Tables 2 or 3).

3. Convergence analysis

In this section, we use the so-called method of ordinary differential equation (ODE) to investigate the convergence behavior of the new SVD-updating algorithms SVD1 and SVD2 derived in the previous section. This method is now commonly used for the convergence analysis of various stochastic recursive algorithms and is described in many advanced textbooks; here, we follow[7].

3.1. Overview of the ODE method

The ODE method is a systematic tool for studying the convergence properties of a wide class of recursive algorithms in a probabilistic sense. The basic principle of this method consists in mapping the algorithm under study into a continuous-time deterministic ODE which describes the expected convergence dynamics of the algorithm. That is, under appropriate conditions, the expected trajectories of the sequence of estimates produced by the recursive algorithm follow the trajectories (i.e. the solutions) of the ODE.

The study of the convergence then amounts to a classical analysis of the algorithm's ODE. Since we are interested here in the asymptotic behavior of algorithms SVD1 and SVD2 as $k \to \infty$, the main focus of the analysis is the study of the ODE's attractors and associated attraction domains. In particular, convergence will be guaranteed, in a sense prescribed by the theory, if we can show that the ODE has a single stable attractor corresponding to the desired solution. Here, Lyapunov stability theory is used to carry out this investigation.

3.2. Signal model

The first step in the application of the ODE method is the formulation of an adequate probalistic model for the sequence of input data driving the recursive algorithm under study. In the case of the algorithms SVD1 and SVD2, this is the sequence of data vectors $\mathbf{x}_k$, $k = 1, 2, ...$

For the purpose of this paper, we shall model the sequence $\{\mathbf{x}_k\}$ as a stationary, temporally white, vector ran-

Operation
for $i = 1 : r$
for $j = i + 1 : r + 1$
$\theta \leftarrow y_i y_j / (\sigma_j^2 - \sigma_i^2)$
$U \leftarrow U G_{ij}(\theta)$
end
end

Table 2. Rsv updating based on V1.

Operation
$z_r \leftarrow y_{r+1}$
for $i = r : -1 : 2$
$\phi_i \leftarrow -\arctan(z_i / y_i)$
$\theta \leftarrow -y_i z_i / \sigma_i^2$
$z_{i-1} \leftarrow \sqrt{y_i^2 + z_i^2}$
$U \leftarrow U G_{i,i+1}(\phi_i + \theta)$
end
$\theta \leftarrow -y_1 z_1 / \sigma_1^2$
$U \leftarrow U G_{1,2}(\theta)$
for $i = 2 : r$
$U \leftarrow U G_{i,i+1}^T(\phi_i)$
end

Table 3. Rsv updating based on V2.

dom process with zero-mean and true covariance matrix

$$R^o = E[\mathbf{x}_k \mathbf{x}_k^H]. \tag{23}$$

This is not the most general model for which the conclusions of this study remain valid, but it has the advantage of simplifying the discussion while preserving the essential aspects of the analysis (a more general model can be found in [7]).

We shall denote by λ_i^o the eigenvalues of R^o and by $\mathbf{u}_i^o$ the corresponding orthonormal eigenvectors, so that

$$R^o = U^o \Lambda^o U^{oH} \tag{24}$$

where $U^o = [\mathbf{u}_1^o, ..., \mathbf{u}_N^o]$, with $U^{oH} U^o = I$, and $\Lambda^o = \text{diag}(\lambda_1^o, ..., \lambda_N^o)$. To simplify the presentation, we shall assume that $\lambda_1^o > ... > \lambda_{r+1}^o = ... = \lambda_N^0$. However, some generalizations are possible.

3.3. Generic form of the algorithms

While the precise mathematical description of the algorithms SVD1 and SVD2 given in Table 1-3 is useful (and necessary) for their practical implementation, it is not the most appropriate one for the application of the ODE method. Indeed, the derivation of the ODE associated to a recursive algorithm generally assumes that the latter has been expressed in the following generic form:

$$\phi_k = \phi_{k-1} + \gamma_k H(\phi_{k-1}, \mathbf{x}_k) + O(\gamma_k^2) \tag{25}$$

where ϕ_k represents the sequence of parameter estimates recursively produced by the algorithm; γ_k is a sequence of small scalar gains; $H(\phi_{k-1}, \mathbf{x}_k)$ is a function which defines, up to first order terms in γ_k, how the parameter ϕ_{k-1} is updated by the algorithms; and $O(\gamma_{k-1}^2)$ represents the residual second and higher-order terms.

Here, the parameter estimates produced by the algorithms at each iteration are the dominant sv $\sigma_{1,k}, ..., \sigma_{r,k}$, the noise subspace singular value $\sigma_{r+1,k}$, and the matrix of dominant rsv $U_{s,k}$. In the following developments, it is more convenient to work with the eigenvalue estimates, i.e. $\lambda_{i,k} = \sigma_{i,k}^2$, than with the sv estimates. Thus, we finally set

$$\phi_k = [U_{s,k}, \lambda_{1,k}, ..., \lambda_{r+1,k}]. \tag{26}$$

In the same way, an arbitrary point in the parameter space is represented by $\phi = [U_s, \lambda_1, ..., \lambda_{r+1}]$.

The gain parameter γ_k is defined as

$$\gamma_k = \beta_k^2 = 1 - \alpha_k^2 \tag{27}$$

where β_k and α_k represent time-varying versions of the fixed parameters β and α appearing in (1). We note that in practice, β is a small parameter, so that the interpretation of γ_k as a sequence of small gain is practically justified.

According to (25), the function $H(\phi, \mathbf{x})$ can be obtained from a linearization of the algorithms in Table 1-3 with respect to the gain parameter γ_k. In this respect, some comments related to the use of the matrix D are necessary. Recall that the latter was introduced to make the components of $\mathbf{y}$ (12) real so as to further simplify the deflated problem. However, the matrix D has no effect on the information contents of the recursive SVD estimates and may actually be omitted if appropriate modifications are made to the algorithms (so as to allow the components of $\mathbf{y}$ and the rotations parameters θ_{ij} (19) to be complex valued).

Due to lack of space, we omit the details of the linearization step and present only the main results. Observing first from (25) that $H(\phi, \mathbf{x})$ belongs to the same space as ϕ_k and is thus structured as in (26), we let $H(.) = [H_{U_s}(.), H_{\lambda_1}(.), ..., H_{\lambda_{r+1}}(.)]$. With this notation, the results of the linearization can be stated as this:

$$H_{\lambda_i}(\phi, \mathbf{x}_k) = \mathbf{e}_i^T U_s^H \mathbf{x}_k \mathbf{x}_k^H U_s \mathbf{e}_i - \lambda_i, \quad i = 1, ..., r, \tag{28}$$

$$H_{\lambda_{r+1}}(\phi, \mathbf{x}_k) = \frac{1}{N-r} \mathbf{x}_k^H (I - U_s U_s^H) \mathbf{x}_k^H - \lambda_{r+1}, \tag{29}$$

$$H_{U_s}(\phi, \mathbf{x}_k) = U_s \Theta - (I - U_s U_s^H) \mathbf{x}_k \mathbf{x}_k^H U_s L^{-1} \tag{30}$$

In these expressions: $\mathbf{e}_i$ is the ith unit vector, $L = \text{diag}(l_{1,r+1}, ..., l_{r,r+1})$, and Θ is a complex $r \times r$ skew-Hermitian matrix (i.e., $\Theta^H = -\Theta$) with entries

$$\theta_{ij} = \begin{cases} \mathbf{e}_i^T U_s^H \mathbf{x}_k \mathbf{x}_k^H U_s \mathbf{e}_j / l_{ij}, & i < j \\ 0, & i = j \end{cases} \tag{31}$$

The algorithm dependent parameters l_{ij} are given by

$$l_{ij} = \begin{cases} \lambda_j - \lambda_i, & \text{for SVD1,} \\ -\lambda_i, & \text{for SVD2.} \end{cases} \quad (32)$$

3.4. The continuous-time ODE

Once a recursive algorithm has been expressed in the generic form (25), the continuous-time ODE describing its expected dynamics can be obtained as

$$\dot{\phi}(\tau) = h(\phi(\tau)) \quad (33)$$

where τ is the fictitious continuous-time variable, $\phi(\tau)$ is a differentiable function of τ representing the expected trajectory of the stochastic recursive algorithm, the dot operator denotes time-derivative and here, $h(\phi) = E[H(\phi, \mathbf{x}_k)]$. The connection between the discrete-time k and the fictitious time τ is achieved via the relation $\tau_k = \sum_{l=1}^{k} \gamma_l$.

In the present application of the ODE method, we let $\phi(\tau) = [U_s(\tau), \lambda_1(\tau), ..., \lambda_{r+1}(\tau)]$, so that the ODE can be expressed as the coupled system

$$\dot{\lambda}_i(\tau) = h_{\lambda_i}(\phi(\tau)), \quad i = 1, ..., r+1, \quad (34)$$

$$\dot{U}_s(\tau) = h_{U_s}(\phi(\tau)). \quad (35)$$

Specification of the ODE associated to the algorithms SVD1 and SVD2 then amounts to a computation of the expected values of the components of the function $H(\phi, \mathbf{x}_k)$, as given in (28)-(30). The results can be summarized as follows:

$$h_{\lambda_i}(\phi) = \mathbf{e}_i^T U_s^H R^o U_s \mathbf{e}_i - \lambda_i, \quad i = 1, ..., r, \quad (36)$$

$$h_{\lambda_{r+1}}(\phi) = \frac{1}{N-r} \mathrm{tr}[(I - U_s U_s^H) R^o] - \lambda_{r+1}, \quad (37)$$

$$h_{U_s}(\phi) = U_s \Psi - (I - U_s U_s^H) R^o U_s L^{-1} \quad (38)$$

where $\Psi = E[\Theta] = [\psi_{ij}]$. From (31), we obtain ($i < j$) $\psi_{ij} = \mathbf{e}_i^T U_s^H R^o U_s \mathbf{e}_j / l_{ij}$.

3.5. Attractors and domains of attraction

We begin by identifying some basic stationary points of the ODE system (34)-(35):

Theorem 1: Let π_i denote an arbitrary permutation of the numbers $i = 1, ..., N$. Let $\mathbf{u}_i$, $i = 1, ..., r$, be an orthonormal basis of eigenvectors of R^o with corresponding eigenvalues $\lambda_i = \lambda_{\pi_i}^o$, and let $\lambda_{r+1} = \frac{1}{(N-r)} \sum_{i=r+1}^{N} \lambda_{\pi_i}^o$. Then $\phi = [\mathbf{u}_1, ..., \mathbf{u}_r, \lambda_1, ..., \lambda_{r+1}]$ is a stationary point of the ODE system (34)-(35), i.e.. $h(\phi) = 0$.

Any attractor ϕ of the ODE system (34)-(35) must be a stationary point of this system, so that Theorem 1 merely provides some candidate attractors. Among these candidates, those corresponding to the trivial permutation $\pi_i = i$ actually represent the desired limit set of the sequence of estimates ϕ_k produced by the algorithms SVD1 and SVD2. We shall denote the set of all stationary points corresponding to this trivial permutation by $\mathcal{D}_*$.

To prove convergence of the algorithms SVD1 and SVD2, we shall show that $\mathcal{D}_*$ is the unique stable attractor of the ODE system (34)-(35), provided some restrictions are imposed on the initial condition $\phi(0)$, as suggested by the following theorem:

Theorem 2: Consider the following manifold of the parameter space: $\mathcal{M} = \{\phi : U_s^H U_s = I\}$. Let $\phi(\tau)$ be a solution of the ODE system (34)-(35). If $\phi(0) \in \mathcal{M}$, then $\phi(\tau) \in \mathcal{M}$ for all $\tau \geq 0$.

In other words, if a trajectory of (34)-(35) originates on $\mathcal{M}$, then it stays on $\mathcal{M}$ at all time. Since the initial conditions for the algorithms SVD1 and SVD2 belong to $\mathcal{M}$ and since the desired solution set $\mathcal{D}_*$ is also included in $\mathcal{M}$, we may thus assume that $\phi(\tau) \in \mathcal{M}$ for all $\tau \geq 0$ in our demonstration that $\mathcal{D}_*$ is a unique stable attractor. Limiting our considerations to trajectories in $\mathcal{M}$ yields the useful identity $\dot{U}_s^H U_s + U_s^H \dot{U}_s = 0$.

Next, we introduce the following Lyapunov function:

$$V(\phi) = \|R(\phi) - R^o\|_F^2 \quad (39)$$

$$R(\phi) = U_s \Lambda_s U_s^H + \lambda_{r+1}(I - U_s U_s^H) \quad (40)$$

where $\Lambda_s = \mathrm{diag}(\lambda_1, ..., \lambda_r)$. One can verify that $V(\phi) = 0$ for any $\phi \in \mathcal{D}_*$. Therefore:

Theorem 3: Any point ϕ in the desired solution set $\mathcal{D}_*$ is a global minimum of $V(\phi)$.

To complete the analysis, we must study the time-derivative of the Lyapunov function $V(\tau) \equiv V(\phi(\tau))$ along an arbitrary trajectory $\phi(\tau)$ of the ODE, subject to the restriction that $\phi(\tau) \in \mathcal{M}$. A lengthy derivation yields:

Theorem 4: In the case of SVD1, $\dot{V}(\tau) \leq 0$ for all $\tau \geq 0$; for SVD2, the same conclusion holds provided $\lambda_1(\tau) > ... > \lambda_{r+1}(\tau)$ for all $\tau \geq 0$.

In the case of SVD1, the theorem implies that all trajectories originating in $\mathcal{M}$ converge to the global minimum $\mathcal{D}_*$, which is then the unique stable attractor of the ODE (34)-(35). In the case of SVD2, we must further require that the sv remain in decreasing order on the trajectory $\phi(\tau)$. In practice, this does not pose a real difficulty since the sv can be rearranged after each update (see step 2-d in Table 1).

4. Illustrative Examples

The performance of the new algorithms SVD1 and SVD2 in subspace tracking applications is investigated via computer experiments. To this end, a conventional model of the type $\mathbf{x}_k = A\mathbf{s}_k + \mathbf{n}_k$ is used, where $\mathbf{s}_k$ is an r-dimensional source vector process, A is a transmission matrix and $\mathbf{n}_k$ is a noise process. The components of the vectors $\mathbf{s}_k$ and $\mathbf{n}_k$

are generated as independent random variables with complex circular Gaussian pdf. The performance of the algorithms is evaluated in terms of the following measures: the rsv error, defined as the distance between the true (i.e. based on R^o (24)) and estimated signal subspaces; a normalized sv error, and when appropriate, the root-MUSIC frequency estimates or their average squared errors. In all cases, exact SVD (SVDe) of the data matrix X_k (1) is used as a benchmark for comparison; N is set to 10.

Typical results for the initial convergence of the algorithms SVD1 and SVD2 are shown in Fig. 1 ($r = 4$, true frequencies $\omega \in \{0, 0.25, 1.0, 1.25\}$, SNR = 15dB, $\beta^2 = .025$, 40 run average). The performance of both SVD1 and SVD2 is comparable to SVDe. Here, the true sv are not particularly well separated (25.46, 21.70, 10.63, 6.16 and 1), which confirms the robustness of SVD2. The effect of a sudden 90^o subspace rotation after convergence is illustrated in Fig. 2 ($r = 2$, $\beta^2 = .1$, 10 run average; see [1] for more details). Finally, the ability of the new algorithms to track narrow-band plane wave sources is illustrated in Fig. 3, which shows the true frequencies and the corresponding root-MUSIC estimates for SVDe, SVD1 and SVD2 ($r = 4$, SNR = dB, $\beta^2 = .025$, single run).

References

[1] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol 78. pp. 1327–1343, Aug. 1990.

[2] M. Moonen, P. V. Dooren and J. Vandewalle, "A singular value decomposition updating algorithm for subspace tracking," *SIAM J. Matrix Anal. Appl.*, vol. 13, pp. 1015–1038, Oct. 1992.

[3] Z. Fu and E. M. Dowling, "Conjugate gradient eigenstructure tracking for adaptive spectral estimation," *IEEE Trans. Signal Processing*, vol. 43, pp. 1151–1160, May 1995.

[4] B. Champagne and Q.-G. Liu, "A new family of EVD tracking algorithms using Givens rotations," in *Proc. IEEE Int. Conf. ASSP*, 1996, pp. V-2539–2542.

[5] I. Karasalo, "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 34, pp. 8–12, Feb. 1986.

[6] B. Champagne, "Adaptive eigendecomposition of data covariance matrices based on first-order perturbations", *IEEE Trans. on Signal Processing*, vol. 42, pp. 2758–2770, Oct. 1994.

[7] A. Benveniste, M. Métivier and P. Priouret, *Adaptive algorithm and Stochastic Approximations*, New York: Springer-Verlag, 1990.

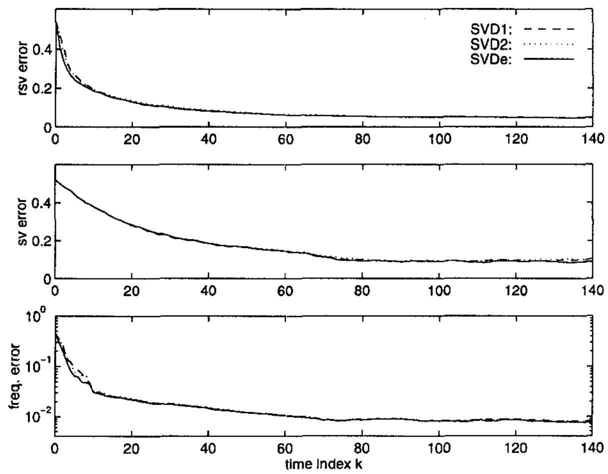

Figure 1. Initial convergence.

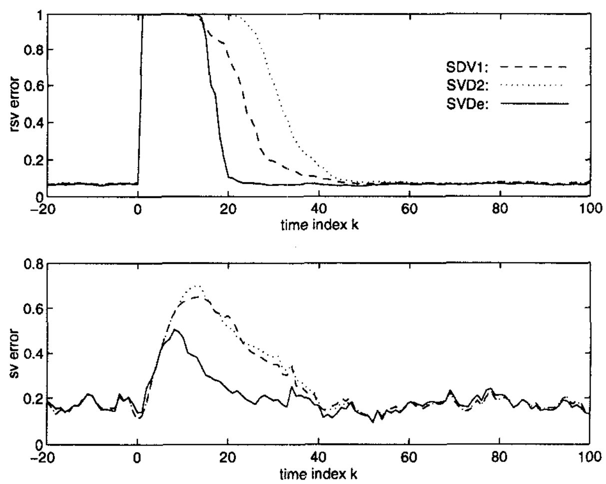

Figure 2. Effect of sudden subspace rotation.

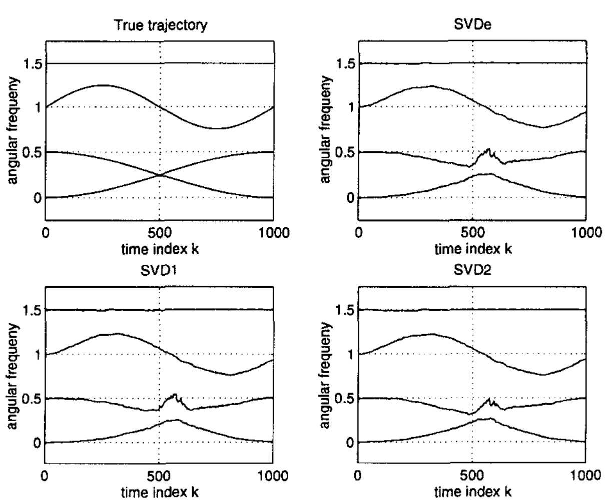

Figure 3. Tracking demonstration.