

# A Constrained Optimization Approach for an Adaptive Generalized Subspace Tracking Algorithm

Amir Valizadeh  
valizadeh@shirazu.ac.ir

## ABSTRACT

In this paper, we present a new algorithm for tracking the generalized signal subspace recursively. It is based on an interpretation of the generalized signal subspace as the solution of a constrained minimization task. This algorithm, referred to as the CGST algorithm, guarantees the  $\mathbf{C}_x$ -orthonormality of the estimated generalized signal subspace basis at each iteration which  $\mathbf{C}_x$  denotes the correlation matrix of the sequence  $\mathbf{x}(t)$ . Thus, the proposed algorithm avoids  $\mathbf{C}_x$ -orthonormalization process after each update for post-processing algorithms which need  $\mathbf{C}_x$ -orthonormal basis for the generalized signal subspace. An efficient implementation of the proposed algorithm enhances applicability of it in real time applications. Various simulation results show outstanding performance of the proposed algorithm.

**Index Terms**— generalized subspace, projection approximation, generalized eigen-decomposition, constrained optimization.

## I. INTRODUCTION

Generalized eigen-decomposition (GED) has an important role in various signal processing applications. Pattern recognition, adaptive noise cancellation and antenna array processing are some of applications of the GED [1]-[3]. Many researches have been focused on principal component analysis (PCA) and signal subspace tracking. Several adaptive algorithms for PCA and signal subspace tracking can be found in the literatures [4]-[7]. Nevertheless, there are a few efficient algorithms for adaptive implementation of GED.

Consider the matrix pencil  $(\mathbf{C}_y, \mathbf{C}_x)$ , where  $\mathbf{C}_y$  and  $\mathbf{C}_x$  are  $N \times N$  symmetric positive definite matrices. Then, the task of computing an  $N \times 1$  vector  $\mathbf{w}$  and a scalar  $\lambda$  such that

$$\mathbf{C}_y \mathbf{w} = \lambda \mathbf{C}_x \mathbf{w} \quad (1)$$

is called the generalized eigen-decomposition problem. The solution vector  $\mathbf{w}$  and scalar  $\lambda$  are called the generalized eigenvector and eigenvalue, respectively, of the matrix pencil  $(\mathbf{C}_y, \mathbf{C}_x)$ . According to the matrix theory, the matrix pencil has  $N$  positive generalized eigenvalues,  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ , and corresponding  $\mathbf{C}_x$ -orthonormal generalized eigenvectors  $\mathbf{v}_i, i = 1, 2, \dots, N$

$$\mathbf{C}_y \mathbf{v}_i = \lambda \mathbf{C}_x \mathbf{v}_i \quad (2)$$

$$\mathbf{v}_i^H \mathbf{C}_x \mathbf{v}_j = \delta_{ij} \quad i, j \in \{1, 2, \dots, N\} \quad (3)$$

where  $\delta_{ij}$  is the Kronecker delta function.

Finding the  $r$  principal generalized eigenvectors of the matrix pencil corresponding to the  $r$  largest generalized eigenvalues is the task of generalized eigen-decomposition. The subspace spanned by the  $r$  principal generalized eigenvectors is called generalized signal subspace. The remaining generalized eigenvectors corresponding to the  $N-r$  generalized eigenvalues span the generalized noise subspace.

Many researchers have addressed the problem (1) for given  $\mathbf{C}_y$  and  $\mathbf{C}_x$  and proposed methods for solving it. Moler and Stewart [8] proposed a QZ algorithm, and Kaufman [9] proposed an LZ algorithm for solving it iteratively. By invoking the Cholesky factorization of  $\mathbf{C}_x$ , the problem (1) can be reduced to the standard eigenvalue problem which has been reported by Martin and Wilkinson [10]. Shougen and Shuqin [11] proposed an algorithm that makes use of Cholesky, QR, and singular value decompositions when  $\mathbf{C}_y$  is also positive definite. Auchmuty [12] proposed and analyzed certain cost functions that are minimized at the eigenvectors corresponding to some specific eigenvalue. These algorithms assume that  $\mathbf{C}_y$  and  $\mathbf{C}_x$  are known, but in adaptive signal processing applications the matrix  $\mathbf{C}_y$  and  $\mathbf{C}_x$  are time-variant and should be estimated from the samples. Thus, these algorithms based on matrix pencil are not feasible for real time signal processing. Most proposed algorithms in literatures [13] are gradient based which slow convergence is the major problem of them.

In this paper, we present a recursive algorithm for tracking the generalized signal subspace of matrix pencil. This algorithm relies on an interpretation of the generalized signal subspace as the solution of a constrained optimization problem. The proposed algorithm employs the projection approximation for obtaining an efficient algorithm. The  $C_x$ -orthonormality of the generalized signal subspace is the constraint used in this optimization. It is noteworthy that the  $C_x$ -orthonormality of the principal generalized signal subspace can improve the performance of the estimation. In addition, it can enhance the convergence rate of the algorithm. We call our approach as constrained generalized subspace tracking (CGST). We will show that the proposed algorithm has  $O(n^2)$  complexity, and thus is appropriate for real time applications. It is noteworthy that in this paper operation counts are expressed in terms of multiply/accumulate (MAC) operations.

This paper is organized as follows. In section II, our approach as a constrained optimization problem is introduced and derivation of the solution is described. Recursive implementation of the proposed solution is derived in section III. In section IV, simulations are used to evaluate the performance of the proposed algorithm and to compare this performance with another existing subspace tracking algorithm. Finally, the main conclusions of this paper are summarized in section V.

## II. CONSTRAINED OPTIMIZATION FOR GENERALIZED SIGNAL SUBSPACE TRACKING

A well-known method for computing the signal subspace of the data is projection approximation subspace tracking (PAST) method [14]. It tracks the dominant subspace of dimension  $r$  spanned by the correlation matrix. Yang *et. al.* [15] proposed an algorithm for generalized eigenvector tracking which invoked the projection approximation. The columns of generalized eigenvector of Yang's method are not exactly  $C_x$ -orthonormal. The deviation from the orthonormality depends on the signal to noise ratio (SNR) and the forgetting factor  $\beta$ . This lack of orthonormality affects seriously the performance of post-processing algorithms which are dependant on  $C_x$ -orthonormality of the generalized basis. To overcome this problem, we propose the following constrained optimization problem.

Let  $\mathbf{x} \in C^n$  and  $\mathbf{y} \in C^n$  be two stationary complex valued random vector processes with the autocorrelation matrices  $C_x = E\{\mathbf{x}\mathbf{x}^H\}$  and  $C_y = E\{\mathbf{y}\mathbf{y}^H\}$  which are assumed to be positive definite. We consider the following minimization problem

$$\begin{aligned} \underset{\mathbf{W}}{\text{minimize}} \quad & J'(\mathbf{W}(t)) = \sum_{i=1}^t \beta^{t-i} [(\mathbf{y}^H(i)\mathbf{C}_x^{-1} - \mathbf{z}^H(i)\mathbf{W}^H(t)) \\ & \times \mathbf{C}_x(\mathbf{C}_x^{-1}\mathbf{y}(i) - \mathbf{W}(t)\mathbf{z}(i))] \\ \text{subject to} \quad & \mathbf{W}^H(t)\mathbf{C}_x\mathbf{W}(t) = \mathbf{I}_r \end{aligned} \quad (4)$$

where  $\mathbf{I}_r$  is the  $r \times r$  identity matrix,  $\mathbf{z}(t) = \mathbf{W}^H(t-1)\mathbf{y}(t)$  is the  $r$ -dimensional compressed data vector, and  $\mathbf{W}$  is an  $n \times r$  ( $r \leq n$ )  $C_x$ -orthonormal generalized subspace basis full rank matrix. The aforementioned constraint guarantees the  $C_x$ -orthonormality of the generalized signal subspace. The use of the forgetting factor  $0 < \beta \leq 1$  is intended to ensure that data in the distant times are downweighted in order to preserve the tracking capability when the system operates in a nonstationary environment.

To solve this constrained problem, we use Lagrange multipliers method. Thus, after expanding the expression for  $J'(\mathbf{W}(t))$ , we can replace (4) with the following problem

$$\begin{aligned} \underset{\mathbf{W}}{\text{minimize}} \quad & h(\mathbf{W}) = \alpha - 2\text{tr}\left(\sum_{i=1}^t \beta^{t-i} \mathbf{y}(i)\mathbf{z}^H(i)\mathbf{W}^H(t)\right) + \\ & \text{tr}\left(\sum_{i=1}^t \beta^{t-i} \mathbf{C}_x\mathbf{W}(t)\mathbf{z}(i)\mathbf{z}^H(i)\mathbf{W}^H(t)\right) + \lambda \left\| \mathbf{W}^H\mathbf{C}_x\mathbf{W} - \mathbf{I}_r \right\|_F^2 \end{aligned} \quad (5)$$

where

$$\alpha = \text{tr}\left(\sum_{i=1}^t \beta^{t-i} \mathbf{y}(i)\mathbf{y}^H(i)\mathbf{C}_x^{-1}\right) \quad (6)$$

which  $\text{tr}(\mathbf{C})$  is the trace of the matrix  $\mathbf{C}$ ,  $\|\cdot\|_F$  denotes the Frobenius norm, and  $\lambda$  is the Lagrange multiplier. We can rewrite  $h(\mathbf{W})$  in the following form

$$\begin{aligned} h(\mathbf{W}) = & \alpha - 2\text{tr}\left(\sum_{i=1}^t \beta^{t-i} \mathbf{y}(i)\mathbf{z}^H(i)\mathbf{W}^H(t)\right) + \\ & \text{tr}\left(\sum_{i=1}^t \beta^{t-i} \mathbf{C}_x\mathbf{W}(t)\mathbf{z}(i)\mathbf{z}^H(i)\mathbf{W}^H(t)\right) + \\ & \lambda \text{tr}(\mathbf{W}^H(t)\mathbf{C}_x\mathbf{W}(t)\mathbf{W}^H(t)\mathbf{C}_x\mathbf{W}(t) - \\ & 2\mathbf{W}^H(t)\mathbf{C}_x\mathbf{W}(t) + \mathbf{I}_r) \end{aligned} \quad (7)$$

Let  $\nabla h = \mathbf{0}$ , where  $\nabla$  is the gradient operator with respect to  $\mathbf{W}$ , then we have

$$\begin{aligned} -\left(\sum_{i=1}^t \beta^{t-i} \mathbf{y}(i)\mathbf{z}^H(i)\right) + \left(\sum_{i=1}^t \beta^{t-i} \mathbf{C}_x\mathbf{W}(t)\mathbf{z}(i)\mathbf{z}^H(i)\right) + \\ \lambda [2\mathbf{C}_x\mathbf{W}(t)\mathbf{W}^H(t)\mathbf{C}_x\mathbf{W}(t) - 2\mathbf{C}_x\mathbf{W}(t) + \mathbf{I}_r] = \mathbf{0} \end{aligned} \quad (8)$$

which can be rewritten in the following form

$$\mathbf{W}(t) = \mathbf{C}_x^{-1} \left( \sum_{i=1}^t \beta^{t-i} \mathbf{y}(i) \mathbf{z}^H(i) \right) \times$$

$$\left[ \sum_{i=1}^t \beta^{t-i} \mathbf{z}(i) \mathbf{z}^H(i) - 2\lambda \mathbf{I}_r + 2\lambda \mathbf{W}^H(t) \mathbf{C}_x \mathbf{W}(t) \right]^{-1} \quad (9)$$

If we substitute  $\mathbf{W}(t)$  from (9) into the constraint which is  $\mathbf{W}^H \mathbf{C}_x \mathbf{W} = \mathbf{I}_r$ , after some manipulations we obtain

$$\left[ \sum_{i=1}^t \beta^{t-i} \mathbf{z}(i) \mathbf{y}^H(i) \right] \mathbf{C}_x^{-1} \left[ \sum_{i=1}^t \beta^{t-i} \mathbf{y}(i) \mathbf{z}^H(i) \right] =$$

$$\left[ \sum_{i=1}^t \beta^{t-i} \mathbf{z}(i) \mathbf{z}^H(i) - 2\lambda \mathbf{I}_r + 2\lambda \mathbf{W}^H(t) \mathbf{C}_x \mathbf{W}(t) \right]^2 \quad (10)$$

Now, we define matrix  $\mathbf{L}$  as follows

$$\mathbf{L} = \sum_{i=1}^t \beta^{t-i} \mathbf{z}(i) \mathbf{z}^H(i) - 2\lambda \mathbf{I}_r + 2\lambda \mathbf{W}^H(t) \mathbf{C}_x \mathbf{W}(t) \quad (11)$$

It follows from substituting (11) into (10) that

$$\mathbf{L} = \left[ \sum_{i=1}^t \beta^{t-i} \mathbf{z}(i) \mathbf{y}^H(i) \right] \mathbf{C}_x^{-1} \left[ \sum_{i=1}^t \beta^{t-i} \mathbf{y}(i) \mathbf{z}^H(i) \right]^{\frac{1}{2}}$$

$$= \left[ \mathbf{C}_{yz}^H(t) \mathbf{C}_x^{-1} \mathbf{C}_{yz}(t) \right]^{\frac{1}{2}} \quad (12)$$

where

$$\mathbf{C}_{yz}(t) = \sum_{i=1}^t \beta^{t-i} \mathbf{y}(i) \mathbf{z}^H(i) \quad (13)$$

and  $(\cdot)^{1/2}$  denotes the square root of a matrix.

Using (13) and (12), we can rewrite (9) in the following form

$$\mathbf{W}(t) = \mathbf{C}_x^{-1}(t) \mathbf{C}_{yz}(t) \mathbf{L}^{-1} \quad (14)$$

Now, using (12) and (14) we can achieve the following fundamental solution

$$\mathbf{W}(t) = \mathbf{C}_x^{-1}(t) \mathbf{C}_{yz}(t) \left[ \mathbf{C}_{yz}^H(t) \mathbf{C}_x^{-1}(t) \mathbf{C}_{yz}(t) \right]^{\frac{-1}{2}} \quad (15)$$

This constrained generalized subspace tracking (CGST) algorithm guarantees the  $\mathbf{C}_x$ -orthonormality of the columns of  $\mathbf{W}(t)$ .

Recursive computation of the  $n \times r$  matrix  $\mathbf{C}_{yz}(t)$  using (13) requires  $2nr$  operations. The computation of  $\mathbf{W}(t)$  using (15) demands additional  $n^2r + 2nr^2 + 3n^2 + n$  operations. Thus, the closed form solution of the CGST method given by (15) needs  $O(n^2r)$  operations. However, some manipulations can reduce the complexity to  $O(n^2)$ .

### III. FAST ADAPTIVE CGST ALGORITHM

We are interested in computing the dominant generalized subspace spanned by correlation matrices of  $\mathbf{C}_x(t)$  and  $\mathbf{C}_y(t)$ . Expression (15) shows the CGST

algorithm for tracking the generalized signal subspace. In order to reduce the order of complexity, we propose the fast CGST algorithm in this section.

#### A. Recursion for the correlation matrix $\mathbf{C}_y(t)$

Let  $\mathbf{y}(t)$  be a sequence of  $n$ -dimensional data vectors. The correlation matrix  $\mathbf{C}_y(t)$  used for generalized signal subspace estimation can be estimated recursively as follows

$$\mathbf{C}_y(t) = \sum_{i=1}^t \beta^{t-i} \mathbf{y}(i) \mathbf{y}^H(i) = \beta \mathbf{C}_y(t-1) + \mathbf{y}(t) \mathbf{y}^H(t) \quad (16)$$

The expression (16) has achieved by the windowing called exponential window. Indeed, this window tends to smooth the variations of the signal parameters and allows a low complexity update at each time. Thus, it is suitable for slowly changing signals.

#### B. Recursion for the cross correlation matrix $\mathbf{C}_{yz}(t)$

The cross correlation matrix  $\mathbf{C}_{yz}(t)$  can be obtained as

$$\mathbf{C}_{yz}(t) = \mathbf{C}_y(t) \mathbf{W}(t-1) \quad (17)$$

For achieving the recursive form for  $\mathbf{C}_{yz}(t)$  in exponential window case, let us substitute (16) into (17)

$$\mathbf{C}_{yz}(t) = \beta \mathbf{C}_y(t-1) \mathbf{W}(t-1) + \mathbf{y}(t) \mathbf{z}^H(t) \quad (18)$$

By applying projection approximation at time  $t-1$ , (18) can be replaced by the following form

$$\mathbf{C}_{yz}(t) = \beta \mathbf{C}_y(t-1) \mathbf{W}(t-2) + \mathbf{y}(t) \mathbf{z}^H(t) \quad (19)$$

By invoking (17), (19) can be expressed as follows

$$\mathbf{C}_{yz}(t) = \beta \mathbf{C}_{yz}(t-1) + \mathbf{y}(t) \mathbf{z}^H(t) \quad (20)$$

#### C. Recursion for generalized signal subspace $\mathbf{W}(t)$

Now, we want to find a recursion for fast update of signal subspace. Let us write (15) as below

$$\mathbf{W}(t) = \mathbf{C}_x^{-1}(t) \mathbf{C}_{yz}(t) \mathbf{\Phi}(t) \quad (21)$$

where

$$\mathbf{\Phi}(t) = \left[ \mathbf{C}_{yz}^H(t) \mathbf{C}_x^{-1}(t) \mathbf{C}_{yz}(t) \right]^{\frac{-1}{2}} \quad (22)$$

It can be easily shown that (21) leads to

$$\mathbf{C}_{yz}(t) = \mathbf{C}_x(t) \mathbf{W}(t) \mathbf{\Phi}^{-1}(t) \quad (23)$$

Substituting (23) into (20) and right multiplying by  $\mathbf{\Phi}(t)$  and left multiplying by  $\mathbf{C}_x^{-1}(t)$  results the following recursion

$$\mathbf{W}(t) = \beta \mathbf{C}_x^{-1}(t) \mathbf{C}_x(t-1) \mathbf{W}(t-1) \mathbf{\Phi}^{-1}(t-1) \mathbf{\Phi}(t) +$$

$$\mathbf{C}_x^{-1}(t) \mathbf{y}(t) \mathbf{z}^H(t) \mathbf{\Phi}(t) \quad (24)$$

By approximation  $\mathbf{C}_x^{-1}(t) \mathbf{C}_x(t-1) \approx \mathbf{I}_n$ , relation (24) can be replaced with

$$\mathbf{W}(t) = \beta \mathbf{W}(t-1) \mathbf{\Phi}^{-1}(t-1) \mathbf{\Phi}(t) + \mathbf{C}_x^{-1}(t) \mathbf{y}(t) \mathbf{z}^H(t) \mathbf{\Phi}(t) \quad (25)$$

Left multiplying (25) by  $\mathbf{W}^H(t-1)$  and right multiplying it by  $\Phi^{-1}(t)$  leads to

$$\mathbf{W}^H(t-1)\mathbf{W}(t)\Phi^{-1}(t) = \beta\Phi^{-1}(t-1) + \mathbf{W}^H(t-1)\mathbf{C}_x^{-1}(t)\mathbf{y}(t)\mathbf{z}^H(t) \quad (26)$$

By applying the approximation  $\mathbf{W}^H(t-1)\mathbf{W}(t) \approx \mathbf{I}_r$ , then we have

$$\Phi^{-1}(t) = \beta\Phi^{-1}(t-1) + \mathbf{W}^H(t-1)\mathbf{C}_x^{-1}(t)\mathbf{y}(t)\mathbf{z}^H(t) \quad (27)$$

For reduce the complexity further, we apply the matrix inversion lemma to (27). The matrix inversion lemma (MIL) can be written as follows

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1})^{-1}\mathbf{D}\mathbf{A}^{-1} \quad (28)$$

Using matrix inversion lemma leads to (27) replaced with the following expression

$$\Phi(t) = \frac{1}{\beta}\Phi(t-1)(\mathbf{I}_r - \frac{\mathbf{f}(t)\mathbf{g}^H(t)}{\mathbf{g}^H(t)\mathbf{f}(t) + \beta}) \quad (29)$$

where

$$\mathbf{f}(t) = \mathbf{W}^H(t-1)[\mathbf{C}_x^{-1}(t)\mathbf{y}(t)] \quad (30)$$

$$\mathbf{g}(t) = \Phi^H(t-1)\mathbf{z}(t) \quad (31)$$

Using (29) results that

$$\beta\Phi^{-1}(t-1)\Phi(t) = \mathbf{I}_r - \frac{\mathbf{f}(t)\mathbf{g}^H(t)}{\mathbf{g}^H(t)\mathbf{f}(t) + \beta} \quad (32)$$

Substituting (32) into (25) leads to the following expression

$$\mathbf{W}(t) = \mathbf{W}(t-1) - \frac{[\mathbf{W}(t-1)\mathbf{f}(t)]\mathbf{g}^H(t)}{\mathbf{g}^H(t)\mathbf{f}(t) + \beta} + [\mathbf{C}_x^{-1}(t)\mathbf{y}(t)][\mathbf{z}^H(t)\Phi(t)] \quad (33)$$

It should be noted that  $\mathbf{C}_x(t)$  is estimated from sequence vectors  $\mathbf{x}(t)$ . We use the exponentially weighted sample correlation matrix for estimating  $\mathbf{C}_x(t)$ . The recursive update relation for  $\mathbf{C}_x(t)$  is written as

$$\mathbf{C}_x(t) = \beta\mathbf{C}_x(t-1) + \mathbf{x}(t)\mathbf{x}^H(t) \quad (34)$$

Using the MIL results the following efficient recursive expression for updating inverse of  $\mathbf{C}_x(t)$

$$\begin{aligned} \mathbf{C}_x^{-1}(t) &= \frac{1}{\beta}\mathbf{C}_x^{-1}(t-1) - \frac{\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)\mathbf{x}^H(t)\mathbf{C}_x^{-1}(t-1)}{\beta\mathbf{x}^H(t)[\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)] + \beta^2} \\ &= \frac{1}{\beta}\mathbf{C}_x^{-1}(t-1)[\mathbf{I}_n - \mathbf{x}(t)\mathbf{q}^H(t)] \end{aligned} \quad (35)$$

where

$$\mathbf{q}(t) = \frac{[\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)]}{\mathbf{x}^H(t)[\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)] + \beta} \quad (36)$$

**Table 1.** The fast CGST algorithm

$\mathbf{W}(0) = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{0}_{(n-r) \times r} \end{bmatrix}, \mathbf{C}_x^{-1}(0) = \mathbf{I}_n, \Phi(0) = \mathbf{I}_r$
$\mathbf{z}(t) = \mathbf{W}^H(t-1)\mathbf{y}(t)$
$\mathbf{a}(t) = \mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)$
$\mathbf{q}(t) = \frac{\mathbf{a}(t)}{\mathbf{x}^H(t)\mathbf{a}(t) + \beta}$
$\mathbf{C}_x^{-1}(t) = \frac{1}{\beta}[\mathbf{C}_x^{-1}(t-1) - \mathbf{a}(t)\mathbf{q}^H(t)]$
$\mathbf{b}(t) = \mathbf{C}_x^{-1}(t)\mathbf{y}(t)$
$\mathbf{g}(t) = \Phi^H(t-1)\mathbf{z}(t)$
$\mathbf{f}(t) = \mathbf{W}^H(t-1)\mathbf{b}(t)$
$\Phi(t) = \frac{1}{\beta}\Phi(t-1)(\mathbf{I}_r - \frac{\mathbf{f}(t)\mathbf{g}^H(t)}{\mathbf{g}^H(t)\mathbf{f}(t) + \beta})$
$\mathbf{c}(t) = \mathbf{z}^H(t)\Phi(t)$
$\mathbf{W}(t) = \mathbf{W}(t-1) - \frac{[\mathbf{W}(t-1)\mathbf{f}(t)]\mathbf{g}^H(t)}{\mathbf{g}^H(t)\mathbf{f}(t) + \beta} + \mathbf{b}(t)\mathbf{c}(t)$

Thus, the above recursive expression can be used in the relations.

The pseudo-code of the fast CGST algorithm is presented in table 1. The total computational complexity of the fast CGST algorithm is  $4n^2 + 5nr + n + O(r^2)$ .

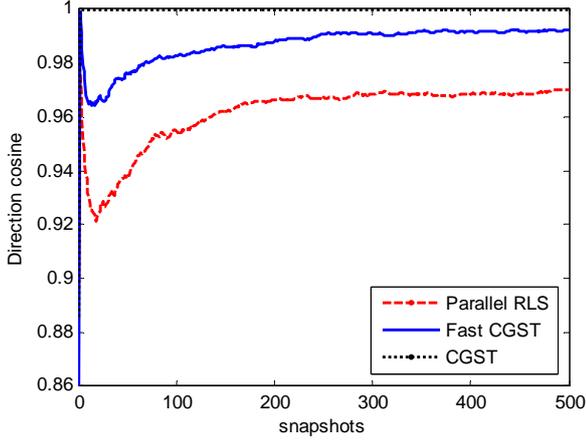
Appropriate initial values should be chosen for  $\mathbf{W}(0)$ ,  $\mathbf{C}(0)$  and  $\Phi(0)$ .  $\mathbf{C}(0)$  and  $\Phi(0)$  must be Hermitian positive definite matrices and  $\mathbf{W}(0)$  should contain  $r$  orthonormal vectors. The choice of these initial values affects the transient behavior but not the steady state behavior of the algorithm. The simplest way is to set  $\mathbf{C}(0)$  and  $\Phi(0)$  to the  $n \times n$  and  $r \times r$  identity matrices and the columns of  $\mathbf{W}(0)$  to the first  $r$  columns of the  $n \times n$  identity matrix.

From computational complexity point of view, the CGST and fast CGST algorithms have a little more complexity than the parallel RLS-based adaptive algorithm. However, the performance of the proposed CGST algorithms is superior to the parallel RLS-based adaptive algorithm.

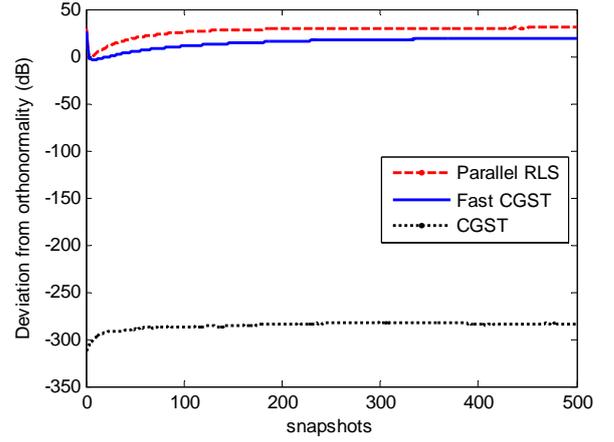
#### IV. SIMULATION RESULTS

In this section, we present some simulation results to demonstrate the performance of the proposed CGST algorithm. In the simulations, we compare the proposed CGST algorithm and fast CGST algorithm with the parallel RLS-based adaptive algorithm [15]. The parallel RLS-based adaptive algorithm has the total computational complexity of  $4n^2 + 3nr + O(r^2)$ .

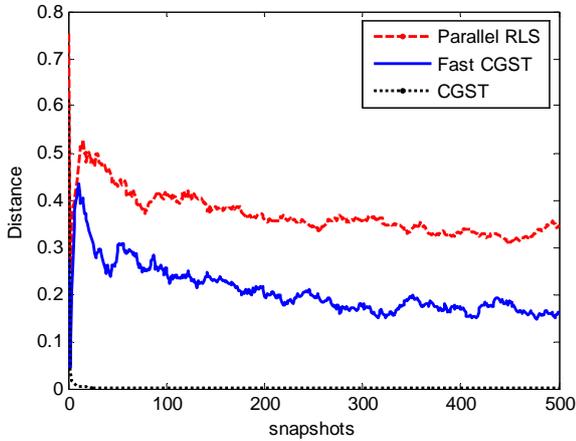
In order to evaluate the convergence speed and the estimated accuracy of the CGST algorithm, the



**Figure 1.** Direction cosine of the first dominant generalized signal subspace for three algorithms



**Figure 3.** Deviation from  $C_x$ -orthonormality of the generalized signal subspace for three algorithms



**Figure 2.** Distance between real and estimated principal generalized signal subspace for three algorithms

direction cosine and subspace distance are respectively defined as

$$\text{direction cosine} = \frac{|\mathbf{w}_i^H \mathbf{v}_i|}{\|\mathbf{w}_i\| \|\mathbf{v}_i\|} \quad (37)$$

$$\text{subspace distance} = \left\| \frac{\mathbf{W}\mathbf{W}^H}{\text{tr}(\mathbf{W}^H \mathbf{W})} - \frac{\mathbf{V}\mathbf{V}^H}{\text{tr}(\mathbf{V}^H \mathbf{V})} \right\|_F \quad (38)$$

where  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r]$  contains the  $r$  dominant generalized eigenvectors of matrix pencil  $(C_y, C_x)$  which can be computed with the direct EVD and  $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_r]$  is the estimated generalized subspace. Zero value of subspace distance shows the identicality of the real and estimated generalized signal subspace. Moreover, when the value of direction cosine approaches to one, the estimated generalized signal subspace converges to the real generalized signal subspace.

In the simulations, the sequences  $x(t)$  and  $y(t)$  are generated by

$$\mathbf{y}(t) = 10 \sin(0.46\pi t) + \mathbf{n}_1(t) \quad (39)$$

$$\mathbf{x}(t) = \sqrt{2} \sin(0.62\pi t) + \mathbf{n}_2(t) \quad (40)$$

where  $\mathbf{n}_1(t)$  and  $\mathbf{n}_2(t)$  are zero mean white Gaussian noises with variances  $\sigma_1^2 = 0.1$  and  $\sigma_2^2 = 0.1$ . The first dominant ( $r=1$ ) generalized subspace of  $(C_y, C_x)$  are considered with the CGST algorithm, fast CGST algorithm, and the parallel RLS-based adaptive algorithm. Let us assume  $N=8$  and  $\beta = 0.99$ . 100 Monte Carlo simulations are performed for each algorithm.

Figure 1 shows the direction cosine of the first dominant generalized signal subspace estimated by algorithms. It can be inferred from this figure that the fast CGST algorithm outperforms the parallel RLS-based algorithm. In addition, the CGST algorithm shows excellent performance.

Figure 2 demonstrates the subspace distance of the algorithms. The fast CGST algorithm shows the better performance than the parallel RLS-based algorithm. The proposed CGST algorithm shows substantial improvement in estimation of the first dominant generalized signal subspace.

The deviation of the subspace weighting matrix  $\mathbf{W}(t)$  from  $C_x$ -orthonormality can be measured by means of the following error criterion

$$\text{deviation} = 20 \log \left( \left\| \mathbf{W}^H(t) \mathbf{C}_x(t) \mathbf{W}(t) - \mathbf{I}_r \right\|_F \right) \quad (41)$$

Figure 3 depicts deviation from orthonormality of the algorithms. It can be easily seen that the performance of the proposed CGST algorithm in the sense of deviation from  $C_x$ -orthonormality is outstanding.

It should be noted that the difference in performance of the CGST and fast CGST algorithms is because of the approximations lead to the expressions of (25) and (27).

## V. CONCLUDING REMARKS

In this paper, we introduced an interpretation of the generalized signal subspace as the solution of a constrained optimization problem. We derived the

solution of this problem and discussed the applicability of the so-called CGST algorithm for tracking the generalized subspace. The complexity of this algorithm is  $O(n^2)$ . Simulations demonstrated excellent performance of the CGST algorithm. In addition, we derived a recursive formulation of this solution for adaptive implementation.

The computational complexity of the fast CGST algorithm is  $O(n^2)$  which is appropriate for on line implementation. The proposed algorithm is efficiently applicable in those post processing applications which need a  $C_x$ -orthonormal basis for the generalized signal subspace.

In order to compare the performance of the proposed CGST and fast CGST algorithm with another subspace tracking algorithm, several simulation scenarios were considered. The simulation results showed that the performance of the CGST algorithms is much better than the parallel RLS-based adaptive algorithm.

## REFERENCES

- [1] L. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey, "Extraction of visual features for lipreading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 198-213, Feb. 2002.
- [2] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*. New York: Wiley, 1996.
- [3] S. Choi, J. Choi, H. J. Im, and B. Choi, "A novel adaptive beamforming algorithm for antenna array CDMA systems with strong interferers," *IEEE Trans. Veh. Tech.*, vol. 51, no. 5, pp. 808-816, Sep. 2002.
- [4] E. Oja, "Principal Components, minor components, and linear neural networks," *Neural Networks*, vol. 5, pp. 927-935, 1992.
- [5] L. Xu, "Least mean square error reconstruction principle for self-organizing neural nets," *Neural Networks*, vol. 6, pp. 627-648, 1993.
- [6] C. E. Davila, "Efficient, high performance, subspace tracking for time-domain data," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3307-3315, Dec. 2000.
- [7] P. Strobach, "Bi-iteration SVD subspace tracking algorithms," *IEEE Trans. Signal Process.*, vol. 45, no. 5, pp. 1222-1240, May 1997.
- [8] C. B. Moler and G. W. Stewart, "An algorithm for generalized matrix eigenvalue problems," *SIAM J. Numerical Anal.*, vol. 10, pp. 241-256, 1973.
- [9] L. Kaufman, "The LZ-algorithm to solve the generalized eigenvalue problem," *SIAM J. Numerical Anal.*, vol. 11, pp. 997-1024, 1974.
- [10] R. S. Msrin and J. H. Wilkinson, "Reduction of the symmetric eigen problem  $\mathbf{Ax}=\lambda\mathbf{Bx}$  and related problems to standard form," *Numerische Mathematik*, vol. 11, pp. 99-110, 1968.
- [11] W. Shougen and Z. Shuqin, "An algorithm for  $\mathbf{Ax}=\lambda\mathbf{Bx}$  with symmetric and positive-definite  $\mathbf{A}$  and  $\mathbf{B}$ ," *SIAM J. Matrix Anal Applications*, vol. 12, no. 4, pp. 654-660, Oct. 1991.
- [12] G. Auchmuty, "Globally and rapidly convergent algorithms for symmetric eigenproblems," *SIAM J. Matrix Anal Applications*, vol. 12, no. 4, pp. 690-706, Oct. 1991.
- [13] C. Chatterjee, V. P. Roychowdhury, J. Ramos, and M. D. Zoltowski, "Self-organizing algorithms for generalized eigen-decomposition," *IEEE Trans. Neural Networks*, vol. 8, no. 6, pp. 1518-1530, Nov. 1997.
- [14] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95-107, Jan. 1995.
- [15] J. Yang, H. Xi, F. Yang, and Y. Zhao, "RLS-based adaptive algorithms for generalized eigen-decomposition," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1177-1188, Apr. 2006.