THE CONSTRAINED GENERALIZED DATA WINDOWING CONJUGATE GRADIENT ALGORITHM

José A. Apolinário Jr.[†] *and Marcello L. R. de Campos*[‡]

[†] *Military Institute of Engineering – Department of Electrical Engineering*
Praça Gen. Tibúrcio, 80 Praia Vermelha, Rio de Janeiro, RJ, Brazil
apolin@ieee.org

[‡] *COPPE/Federal University of Rio de Janeiro – Electrical Engineering Program*
P.O.Box 68504, Rio de Janeiro, RJ, 21941-972, Brazil
mcampos@ieee.org

[‡] *UniK — University of Oslo*
Instituttveien–25, Kjeller, Norway, 2027

ABSTRACT

This paper introduces a constrained version of a recently proposed generalized data windowing scheme applied to the Conjugate Gradient algorithm. This scheme combines two types of data windowing, the finite-data sliding window and the exponentially weighted data window, in an attempt to attain the best of both methods in a linearly constrained scenario. The proposed algorithm was tested in a simple adaptive beamforming application, where the expected better performance was demonstrated.

1. INTRODUCTION

The Conjugate Gradient (CG) method has been successfully employed in adaptive filtering [1]–[5] in an attempt to attain fast speed of convergence with low computational complexity. The CG method, solving the equation $\mathbf{R}\mathbf{w} = \mathbf{p}$ for $\mathbf{w}$, when used in adaptive filtering, needs the estimation of the input-signal correlation matrix $\mathbf{R}(k) = E\left[\mathbf{x}(k)\mathbf{x}^T(k)\right]$ and the cross-correlation vector $\mathbf{p} = E\left[d(k)\mathbf{x}(k)\right]$, $\mathbf{x}(k)$ being the input-signal vector and $d(k)$ the reference or *desired* signal [6]. This estimation may be carried out in different ways, e.g., the finite-data sliding window and the exponentially weighted data window. A recently proposed algorithm combining both schemes was presented in [7], which we refer to in this work as the Generalized Data Windowing Conjugate Gradient (GDWCG) algorithm.

Linearly constrained adaptive filters (LCAF) have found applications in a number of practical problems, including adaptive beamforming with sensor arrays and multi-user detection in mobile communication systems. The CG method was also employed in linearly constrained adaptive filtering. A constrained version of the Modified Conjugate Gradient (MCG) algorithm from [3] was introduced in [8]. The constrained version was equivalent, in infinite precision environment, to the MCG algorithm used within the so-called Generalized Sidelobe Canceler (GSC) structure [9, 10]. This structure is well-known to be able to transform the linearly constrained minimization problem into an unconstrained minimization problem.

In order to minimize the Mean Squared-Error (MSE) with respect to $\mathbf{w}$, subject to $\mathbf{C}^{\mathrm{T}}\mathbf{w} = \mathbf{f}$, the GSC structure decomposes the coefficient vector using a transformation matrix that can be represented by $[\mathbf{C} \quad -\mathbf{B}]$, where $\mathbf{C}$ is the constraint matrix, $\mathbf{w}$ is the coefficient vector, $\mathbf{f}$ is the gain vector, and $\mathbf{B}$ is the blocking matrix which spans the null space of the constraint matrix $\mathbf{C}$, i.e., $\mathbf{B}^{\mathrm{T}}\mathbf{C} = \mathbf{0}$. The transformed coefficient vector is intrinsically partitioned yielding and overall filter $\mathbf{w}(k) = \mathbf{F} - \mathbf{B}\mathbf{w}_{\mathrm{GSC}}(k)$, where $\mathbf{F} = \mathbf{C}(\mathbf{C}^{\mathrm{T}}\mathbf{C})^{-1}\mathbf{f}$ and $\mathbf{w}_{\mathrm{GSC}}$ is the reduced-dimension coefficient vector that operates on the input-signal vector modified by the blocking matrix $\mathbf{B}$.

Given a projection matrix $\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^{\mathrm{T}}\mathbf{C})^{-1}\mathbf{C}^{\mathrm{T}}$, then any constrained adaptive filter $\mathbf{w}(k)$ can be decomposed in two parts: A projection onto the subspace orthogonal to the space spanned by the constraint matrix $\mathbf{C}$, i.e., $\mathbf{w}(k)$ premultiplied by the projection matrix $\mathbf{P}$, and a translation that brings the projected vector back to the hyperplane $\mathbf{C}^{\mathrm{T}}\mathbf{w} = \mathbf{f}$, i.e.,

$$\mathbf{w}(k) = \mathbf{P}\mathbf{w}(k) + \mathbf{F} \tag{1}$$

where $\mathbf{F}$ is as given above and $\mathbf{P}$ is the projection matrix, which can also be written as $\mathbf{P} = \mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}$.

This paper introduces a constrained version of the GDWCG algorithm, namely the CGDWCG algorithm, and is organized as follows: After Section 2 presents the equations of the GDWCG algorithm, Section 3 presents a step-by-step derivation of the new algorithm, and Section 4 presents the simulation results. Finally, Section 5 concludes this work.

This work was supported in part by CNPq (projects 310258/2006-5, 303408/2004-9 and 306445/2007-7), CAPES (project 3644-07-3), and the Research Council of Norway (VERDIKT project *Mobile-to-Mobile Communication Systems - M2M* number 183311/S10).

2. THE GENERALIZED DATA WINDOWING CONJUGATE GRADIENT (GDWCG) ALGORITHM

The objective function used for the conjugate gradient method is usually represented by $J_{\mathbf{w}} = \frac{1}{2}\mathbf{w}^{\mathrm{T}}(k)\mathbf{R}\mathbf{w}(k) - \mathbf{p}^{\mathrm{T}}\mathbf{w}(k)$ such that its gradient with respect to $\mathbf{w}(k)$, when set to zero, yields $\mathbf{R}\mathbf{w}(k) = \mathbf{p}$. A residual vector is defined as $\mathbf{g}(k) = \mathbf{p} - \mathbf{R}\mathbf{w}(k)$ and we update $\mathbf{w}(k)$ in the direction of $\mathbf{c}(k)$, an $\mathbf{R}$-conjugate direction to $\mathbf{c}(k-1)$, such that the squared norm of the residual vector, $\|\mathbf{g}(k)\|^2$, is reduced at each iteration:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \alpha(k)\mathbf{c}(k) \tag{2}$$

where $\alpha(k)$ is a step-size computed to minimize $J_{\mathbf{w}}$.

Taking the partial derivative of $J_{\mathbf{w}}$ with respect to $\alpha(k)$ and setting the result equal to zero, we obtain an expression for $\alpha(k)$ which leads to $\alpha(k) = \frac{\mathbf{g}^{\mathrm{T}}(k-1)\mathbf{c}(k)}{\mathbf{c}^{\mathrm{T}}(k)\mathbf{R}\mathbf{c}(k)}$. A very simple updating expression for the direction vector is obtained by assuming

$$\mathbf{c}(k+1) = \mathbf{g}(k) + \beta(k)\mathbf{c}(k), \tag{3}$$

$\beta(k)$ being obtained from imposing $\mathbf{c}^{\mathrm{T}}(k+1)\mathbf{R}\mathbf{c}(k) = 0$:

$$\beta(k) = -\frac{\mathbf{c}^{\mathrm{T}}(k)\mathbf{R}\mathbf{g}(k)}{\mathbf{c}^{\mathrm{T}}(k)\mathbf{R}\mathbf{c}(k)}. \tag{4}$$

When applying this method to adaptive filters, the correlation matrix $\mathbf{R}$ as well as the cross-correlation vector $\mathbf{p}$ are to be estimated iteratively. This estimation can be carried out according to different windowing schemes, e.g., the ones presented in [3] for the MCG algorithm:

1. Finite-data sliding window: Only the data with a finite length (M) window are used to estimate the correlation matrix $\mathbf{R}(k)$ and the cross-correlation vector $\mathbf{p}(k)$, i.e.,

$$\mathbf{R}(k) = \frac{1}{M}\sum_{j=0}^{M-1}\mathbf{x}(k-j)\mathbf{x}^{\mathrm{T}}(k-j) \tag{5}$$

$$\mathbf{p}(k) = \frac{1}{M}\sum_{j=0}^{M-1} d(k-j)\mathbf{x}(k-j) \tag{6}$$

2. Exponentially weighted data window: The resulting correlation matrix and the cross-correlation vector are the same as the ones used by the conventional exponentially-weighted RLS algorithm, i.e.,

$$\mathbf{R}(k) = \lambda\mathbf{R}(k-1) + \mathbf{x}(k)\mathbf{x}^{\mathrm{T}}(k) \tag{7}$$

$$\mathbf{p}(k) = \lambda\mathbf{p}(k-1) + d(k)\mathbf{x}(k) \tag{8}$$

where λ is a forgetting factor.

As seen in [3] and [7], the use of finite-data sliding windowing, depending on the value of M, may result in high misadjustment and slow convergence (low values of M). On the other hand, the exponentially weighted windowing leads to RLS-compatible misadjustment but the algorithm may suffer from slow convergence in the *degenerated* scheme of the MCG algorithm having a single iteration per coefficient-vector update. In an attempt to combine fast convergence and RLS-like misadjustment with low computational complexity, a generalized data windowing scheme was proposed in [7], which is given below:

$$\mathbf{R}(k) = \lambda\mathbf{R}(k-1) + \frac{1}{M}\sum_{j=0}^{M-1}\mathbf{x}(k-j)\mathbf{x}^{\mathrm{T}}(k-j) \tag{9}$$

$$\mathbf{p}(k) = \lambda\mathbf{p}(k-1) + \frac{1}{M}\sum_{j=0}^{M-1} d(k-j)\mathbf{x}(k-j) \tag{10}$$

In order to reduce the computational complexity, the data correlation matrix can be computed as

$$\mathbf{R}(k) = \lambda\mathbf{R}(k-1) + \widetilde{\mathbf{R}}(k), \tag{11}$$

where

$$\widetilde{\mathbf{R}}(k) = \widetilde{\mathbf{R}}(k-1) + \frac{1}{M}\left[\mathbf{x}(k)\mathbf{x}^{\mathrm{T}}(k) - \mathbf{x}(k-M)\mathbf{x}^{\mathrm{T}}(k-M)\right]. \tag{12}$$

Similarly, the cross-correlation vector is more efficiently computed with

$$\mathbf{p}(k) = \lambda\mathbf{p}(k-1) + \widetilde{\mathbf{p}}(k), \tag{13}$$

where

$$\widetilde{\mathbf{p}}(k) = \widetilde{\mathbf{p}}(k-1) + \frac{1}{M}\left[d(k)\mathbf{x}(k) - d(k-M)\mathbf{x}(k-M)\right]. \tag{14}$$

This scheme resulted in the Generalized Data Windowing Conjugate Gradient (GDWCG) algorithm introduced in [7]. The residual vector $\mathbf{g}(k) = \mathbf{p}(k) - \mathbf{R}(k)\mathbf{w}(k)$, after replacing $\mathbf{p}(k)$ and $\mathbf{R}(k)$ using (11) and (13), takes the form

$$\mathbf{g}(k) = \lambda\mathbf{g}(k-1) - \alpha\mathbf{R}(k)\mathbf{c}(k) + \widetilde{\mathbf{p}}(k) - \widetilde{\mathbf{R}}(k)\mathbf{w}(k-1). \tag{15}$$

For the computation of the step-size $\alpha(k)$, the multiplicative parameter η in [3] was made equal to λ for its proved optimality, as shown in [4] (with correction in [5]). The result is as follows:

$$\alpha(k) = \lambda\frac{\mathbf{c}^{\mathrm{T}}(k)\mathbf{g}(k-1)}{\mathbf{c}^{\mathrm{T}}(k)\mathbf{R}(k)\mathbf{c}(k)}. \tag{16}$$

Finally, completing the equations of the GDWCG algorithm, the (non-reset) Polak-Ribiere method is employed for improved performance [3] in the computation of parameter $\beta(k)$:

$$\beta(k) = \frac{\left[\mathbf{g}(k) - \mathbf{g}(k-1)\right]^{\mathrm{T}}\mathbf{g}(k)}{\mathbf{g}^{\mathrm{T}}(k-1)\mathbf{g}(k-1)}. \tag{17}$$

3. THE CONSTRAINED GDW CONJUGATE GRADIENT ALGORITHM

Based on the general data windowing scheme from [7] and following the same approach used in [8], we start the derivation of a constrained version of this algorithm by employing this windowing strategy in a GSC structure (with orthogonal blocking matrix). The data correlation matrix becomes

$$\begin{aligned}\mathbf{R}_{\text{GSC}}(k) &= \lambda\mathbf{R}_{\text{GSC}}(k-1) + \frac{1}{M}\sum_{j=0}^{M-1}\mathbf{x}_{\text{GSC}}(k-j)\mathbf{x}_{\text{GSC}}^{\text{T}}(k-j)\\ &= \mathbf{B}^{\text{T}}\mathbf{R}(k)\mathbf{B},\end{aligned} \quad (18)$$

where $\mathbf{R}(k)$ is the data-correlation of the overall adaptive filter, as in (9).

The overall coefficient vector is obtained from the GSC structure using the updating expression of the unconstrained GDWCG algorithm:

$$\begin{aligned}\mathbf{w}(k) &= \mathbf{F} - \mathbf{B}\mathbf{w}_{\text{GSC}}(k)\\ &= \mathbf{F} - \mathbf{B}\left[\mathbf{w}_{\text{GSC}}(k-1) + \alpha_{\text{GSC}}(k)\mathbf{c}_{\text{GSC}}(k)\right]\\ &= \mathbf{P}\mathbf{w}(k-1) + \mathbf{F} - \alpha_{\text{GSC}}(k)\mathbf{B}\mathbf{c}_{\text{GSC}}(k).\end{aligned} \quad (19)$$

Notice, in the previous expression, that the possible simplification $\mathbf{P}\mathbf{w}(k-1)+\mathbf{F} = \mathbf{w}(k-1)$ is not used in order to avoid drift from the constraint hyperplane $\mathbf{C}^{\text{T}}\mathbf{w} = \mathbf{f}$ in finite precision implementations [11].

Also in the above expression, the step-size $\alpha_{\text{GSC}}(k)$ and the direction vector $\mathbf{c}_{\text{GSC}}(k)$ are given as

$$\alpha_{\text{GSC}}(k) = \lambda\frac{\mathbf{c}_{\text{GSC}}^{\text{T}}(k)\mathbf{g}_{\text{GSC}}(k-1)}{\mathbf{c}_{\text{GSC}}^{\text{T}}(k)\mathbf{R}_{\text{GSC}}(k)\mathbf{c}_{\text{GSC}}(k)} \quad (20)$$

and

$$\mathbf{c}_{\text{GSC}}(k+1) = \mathbf{g}_{\text{GSC}}(k) + \beta_{\text{GSC}}(k)\mathbf{c}_{\text{GSC}}(k). \quad (21)$$

Completing the equations of the GSC-GDWCG algorithm, we have the following expressions for the residual vector and the parameter $\beta(k)$:

$$\begin{aligned}\mathbf{g}_{\text{GSC}}(k) &= \lambda\mathbf{g}_{\text{GSC}}(k-1) - \alpha_{\text{GSC}}(k)\mathbf{R}_{\text{GSC}}(k)\mathbf{c}_{\text{GSC}}(k)\\ &\quad + \widetilde{\mathbf{p}}_{\text{GSC}}(k) - \widetilde{\mathbf{R}}_{\text{GSC}}(k)\mathbf{w}_{\text{GSC}}(k-1)\end{aligned} \quad (22)$$

and

$$\beta_{\text{GSC}}(k) = \frac{\left[\mathbf{g}_{\text{GSC}}(k) - \mathbf{g}_{\text{GSC}}(k-1)\right]^{\text{T}}\mathbf{g}_{\text{GSC}}(k)}{\mathbf{g}_{\text{GSC}}^{\text{T}}(k-1)\mathbf{g}_{\text{GSC}}(k-1)}. \quad (23)$$

For the constrained version of this algorithm, we make $\mathbf{c}(k) = \mathbf{B}\mathbf{c}_{\text{GSC}}(k)$ and $\mathbf{g}(k) = \mathbf{B}\mathbf{g}_{\text{GSC}}(k)$, such that the two variables $\alpha(k)$ and $\beta(k)$ will be given by the following expressions:

$$\alpha(k) = \alpha_{\text{GSC}}(k) = \lambda\frac{\mathbf{c}^{\text{T}}(k)\mathbf{g}(k-1)}{\mathbf{c}^{\text{T}}(k)\overline{\mathbf{R}}(k)\mathbf{c}(k)} \quad (24)$$

and

$$\beta(k) = \beta_{\text{GSC}}(k) = \frac{\left[\mathbf{g}(k) - \mathbf{g}(k-1)\right]^{\text{T}}\mathbf{g}(k)}{\mathbf{g}^{\text{T}}(k-1)\mathbf{g}(k-1)} \quad (25)$$

where $\overline{\mathbf{R}}(k) = \mathbf{B}\mathbf{R}_{\text{GSC}}(k)\mathbf{B}^{\text{T}} = \mathbf{P}\mathbf{R}(k)\mathbf{P}$, for in our case $\mathbf{B}^{\text{T}}\mathbf{B} = \mathbf{I}$ and hence $\mathbf{P} = \mathbf{B}\mathbf{B}^{\text{T}}$.

The equation of the updated search direction $\mathbf{c}(k+1)$ is then easily found as the product $\mathbf{B}\mathbf{c}_{\text{GSC}}(k+1)$, given by

$$\mathbf{c}(k+1) = \mathbf{g}(k) + \beta(k)\mathbf{c}(k). \quad (26)$$

With the new definitions of $\mathbf{c}(k)$ and $\mathbf{g}(k)$, the updating equation of the constrained algorithm becomes

$$\mathbf{w}(k) = \mathbf{P}\mathbf{w}(k-1) + \mathbf{F} - \alpha(k)\mathbf{c}(k). \quad (27)$$

From (22) and making $\mathbf{g}(k) = \mathbf{B}\mathbf{g}_{\text{GSC}}(k)$, after some algebraic manipulation, the recursive formulation for the residual vector is given as

$$\mathbf{g}(k) = \lambda\mathbf{g}(k-1) - \alpha(k)\overline{\mathbf{R}}(k)\mathbf{c}(k) + \widetilde{\overline{\mathbf{p}}}(k) + \widetilde{\overline{\mathbf{R}}}(k)\mathbf{w}(k-1) \quad (28)$$

where, from (18) and the previous definition of $\overline{\mathbf{R}}(k)$, we have:

$$\overline{\mathbf{R}}(k) = \lambda\overline{\mathbf{R}}(k-1) + \widetilde{\overline{\mathbf{R}}}(k) \quad (29)$$

and

$$\widetilde{\overline{\mathbf{R}}}(k) = \widetilde{\overline{\mathbf{R}}}(k-1) + \frac{1}{M}\left[\overline{\mathbf{x}}(k)\overline{\mathbf{x}}^{\text{T}}(k) - \overline{\mathbf{x}}(k-M)\overline{\mathbf{x}}^{\text{T}}(k-M)\right]. \quad (30)$$

Similarly, defining $\widetilde{\overline{\mathbf{p}}}(k) = \mathbf{B}\widetilde{\mathbf{p}}_{\text{GSC}}(k)$, recalling the definition of $\widetilde{\mathbf{p}}(k)$, taking into account that in a GSC structure we define $d_{\text{GSC}}(k) = \mathbf{F}^{\text{T}}\mathbf{x}(k) - d(k)$, the following update expression for $\widetilde{\overline{\mathbf{p}}}(k)$ is obtained:

$$\widetilde{\overline{\mathbf{p}}}(k) = \widetilde{\overline{\mathbf{p}}}(k-1) + \frac{1}{M}\left[\bar{d}(k)\overline{\mathbf{x}}(k) - \bar{d}(k-M)\overline{\mathbf{x}}(k-M)\right] \quad (31)$$

with $\bar{d}(k) = \mathbf{F}^{\text{T}}\mathbf{x}(k) - d(k)$.

All expressions for the new algorithm, the CGDWCG algorithm in its complex version, are shown in Table 1. Notice that a small positive number δ was introduced in all denominators to avoid division by zero.

4. SIMULATION RESULTS

We evaluated the new CGDWCG algorithm and compared its performance with the CCG algorithm, applying both algorithms to adaptive beamforming. We intended to show improvement in speed of convergence as a consequence of using the combination of sliding windowing and exponentially weighting windowing in the constrained conjugate-gradient algorithm. The CCG algorithm chosen was the one presented in [8] with exponentially weighted estimates of matrices $\mathbf{R}$ and $\mathbf{p}$, which is equivalent to the new CGDWCG algorithm

Table 1. Constrained GDW Conjugate Gradient Algorithm

Initialization:

$0 \ll \lambda \leq 1, \quad \delta$ small positive number

$\mathbf{w}(0) = \mathbf{F} = \mathbf{C}\left(\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)^{-1}\mathbf{f}$

$\widetilde{\overline{\mathbf{R}}}(0) = \mathbf{0}$

$\overline{\mathbf{R}}(0) = \mathbf{P} = \mathbf{I} - \mathbf{C}\left(\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)^{-1}\mathbf{C}^{\mathrm{T}}$

$\widetilde{\overline{\mathbf{p}}}(0) = \mathbf{g}(0) = \mathbf{c}(0) = \mathbf{0}$

for each k

$\{ \quad \overline{\mathbf{x}}(k) = \mathbf{P}\mathbf{x}(k)$

$\widetilde{\overline{\mathbf{R}}}(k) = \widetilde{\overline{\mathbf{R}}}(k-1) + \frac{1}{M}\left[\overline{\mathbf{x}}(k)\overline{\mathbf{x}}^{\mathrm{H}}(k) - \overline{\mathbf{x}}(k-M)\overline{\mathbf{x}}^{\mathrm{H}}(k-M)\right]$

$\bar{d}^*(k)\overline{\mathbf{x}}(k) = \left[\mathbf{F}^{\mathrm{H}}\mathbf{x}(k) - d(k)\right]^*\overline{\mathbf{x}}(k)$

$\widetilde{\overline{\mathbf{p}}}(k) = \widetilde{\overline{\mathbf{p}}}(k-1) + \frac{1}{M}\left[\bar{d}^*(k)\overline{\mathbf{x}}(k) - \bar{d}^*(k-M)\overline{\mathbf{x}}(k-M)\right]$

$\overline{\mathbf{R}}(k) = \lambda\overline{\mathbf{R}}(k-1) + \widetilde{\overline{\mathbf{R}}}(k)$

$\alpha(k) = \lambda\frac{\mathbf{c}^{\mathrm{H}}(k)\mathbf{g}(k-1)}{\mathbf{c}^{\mathrm{H}}(k)\overline{\mathbf{R}}(k)\mathbf{c}(k)+\delta}$

$\mathbf{g}(k) = \lambda\mathbf{g}(k-1) - \alpha(k)\overline{\mathbf{R}}(k)\mathbf{c}(k) + \widetilde{\overline{\mathbf{p}}}(k) + \widetilde{\overline{\mathbf{R}}}(k)\mathbf{w}(k-1)$

$\mathbf{w}(k) = \mathbf{P}\mathbf{w}(k-1) + \mathbf{F} - \alpha(k)\mathbf{c}(k)$

$\beta(k) = \frac{[\mathbf{g}(k)-\mathbf{g}(k-1)]^{\mathrm{H}}\mathbf{g}(k)}{\mathbf{g}^{\mathrm{H}}(k-1)\mathbf{g}(k-1)+\delta}$

$\mathbf{c}(k+1) = \mathbf{g}(k) + \beta(k)\mathbf{c}(k)$

$\}$

presented here when $M = 1$. For both algorithms, the value of λ in the experiment was set to $\lambda = 0.98$, and the experiments were run for ensembles of 10,000 tests. The results shown in the figures are the averages for all tests in the ensemble.

The system setup was that of a typical uniform linear array (ULA) with 7 antennas spaced at half of the wavelength. The received discrete-time input signal was formed by a user of interest with look-direction set as 0^o and three interferers (or jammers) whose impinging angles were 35^o, 45^o, and 50^o. The input-signal model used was

$$\mathbf{x}(k) = \mathbf{S}\mathbf{A}\mathbf{u}(k) + \mathbf{n}(k) \tag{32}$$

where $\mathbf{S} = [\mathbf{s}(\theta_1) \cdots \mathbf{s}(\theta_4)]$ is the steering matrix containing the steering vectors of the users, θ_i is the direction of arrival (DoA) for user i, $\mathbf{A} = \mathrm{diag}[\mathbf{A}_1 \cdots \mathbf{A}_4]$ contains the user amplitudes, $\mathbf{u}(k) = [u_1(k) \cdots u_4(k)]$ contains the transmitted user information, and $\mathbf{n}(k)$ is the sampled noise vector. The signal-to-noise ratio was 0dB and the jammer-to-noise ratios were equal to 30dB for all interferers.

Figures 1 and 2 show a comparison of MSE and coefficient-error vector between the two algorithms: CCG algorithm and CGDWCG algorithm. The curves show the performances of the CGDWCG algorithm for 4 different values of M, i.e., different values of the length of the sliding window: $M = 1$, $M = 5$, $M = 10$, and $M = 20$. The curves for the CCG and CGDWCG algorithms are coincident for $M = 1$, as expected.

We can clearly see that for this application the use of a sliding window combined with an exponentially weighted window helps improving the speed of convergence. Figure 3 compares the beampattern obtained by the two algorithms after 25 samples. It is clear that the slight gain in speed of convergence is a powerful asset if one needs superior performance after only few iterations, for one may see clearly that the beampattern achieved with the CGDWCG algorithm is closer to the optimal beampattern after very few iterations.

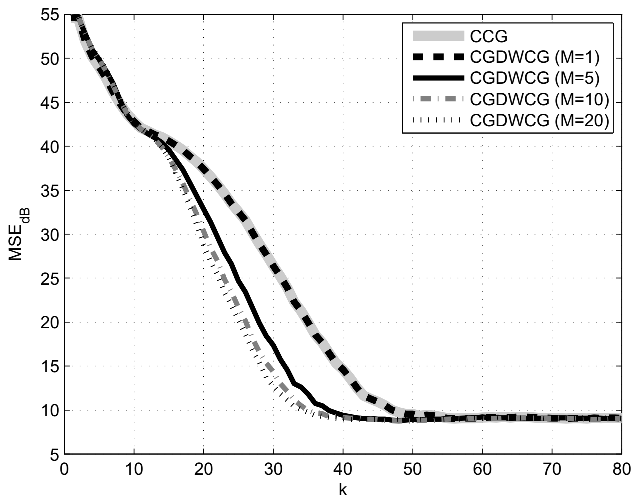

Fig. 1. Mean-squared error.

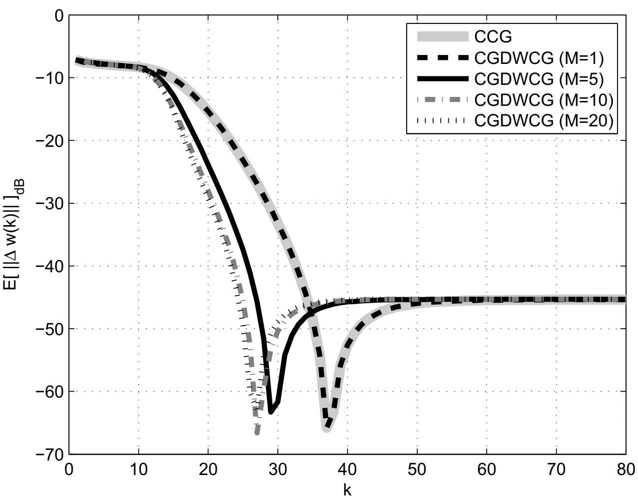

Fig. 2. Norm of coefficient-error vector.

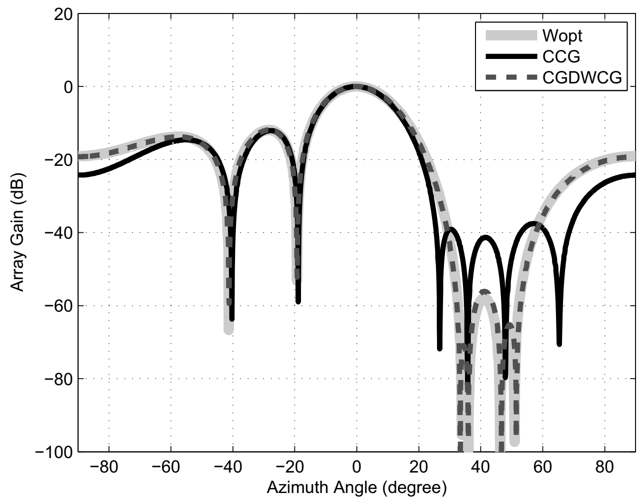

Fig. 3. Beampattern.

Superior performance of the CGDWCG algorithm when compared to the CCG algorithm was more pronounced in this application when interferers were close to each other. For other values of JNR even more significant results could be obtained.

Similar results are also expected in other applications of constrained adaptive filters, such as, e.g., multi-user detection in CDMA mobile communications systems and linear-phase system identification.

The increase in computational complexity due to the introduction of the sliding window is marginal, as can be seen in Table 1 when the value of M is increased from $M = 1$. Besides, computational complexity is not a function of M for $M > 1$, although we have noticed that improvement in performance is marginal as M is further increased. For the simulations shown here, performance improvements ceased for $M > 10$.

5. CONCLUSION

In this article we present the constrained version of the conjugate-gradient algorithm which employs a combination of exponential and sliding windowing for estimating the autocorrelation matrix of the input signal and the cross-correlation vector between the input signal and the desired signal. Algorithm derivation follows closely that of the constrained conjugate-gradient algorithm presented in [8]. The simulation results from an experiment carried out on adaptive beamforming have shown a superior performance of the proposed algorithm in this application when signals from the jammers come form directions having a small difference of DOA. This algorithm, due to its small sample support requirement, can be successfully applied to the field of smart antennas.

6. REFERENCES

[1] G. K. Boray and M. D. Srinath, "Conjugate gradient techniques for adaptive filtering," *IEEE Transactions on Circuits and Systems — I: Fundamental Theory and Applications*, vol. 39, no. 1, pp. 1–10, January 1992.

[2] T. Bose and M.-Q. Chen, "Conjugate gradient method in adaptive bilinear filterin," *IEEE Transactions of Signal Processing*, vol. 43, no. 6, pp. 1503–1508, June 1995.

[3] P. S. Chang and A. N. Willson Jr., "Analysis of conjugate gradient algorithms for adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 48, pp. 409–418, December 2000.

[4] O. Diene and A. Bhaya, "Adaptive filtering algorithms designed using control Liapunov functions," *IEEE Signal Processing Letters*, vol. 13, pp. 224–227, April 2006.

[5] O. Diene and A. Bhaya, "Correction "adaptive filtering algorithms designed using control Liapunov functions"," *IEEE Signal Processing Letters*, vol. 14, pp. 1047, December 2007.

[6] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Kluwer Academic Publishers, second edition, 2002.

[7] Z. Shengkui, M. Zhihong, and K. Suiyang, "Conjugate gradient algorithm design with rls normal equation," in *Proceedings of the International Conference on Information, Communications, and Signal Processing*, 2007.

[8] J. A. Apolinário Jr., M. L. R. de Campos, and C. P. B. Oñate, "The constrained conjugate gradient algorithm," *IEEE Signal Processing Letters*, vol. 7, pp. 351–354, December 2000.

[9] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transactions on Antennas and Propagation*, vol. AP-30, no. 1, pp. 27–34, January 1982.

[10] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, fourth edition, 2002, ISBN-13: 978-0130901262.

[11] O. L. Frost III, "An algorithm for linearly-constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, pp. 926–935, August 1972.