THE CONSTRAINED AFFINE PROJECTION ALGORITHM — DEVELOPMENT AND CONVERGENCE ISSUES

Marcello L. R. de Campos[1] *and José Antonio Apolinário Jr.*[2]

[1]COPPE/Universidade Federal do Rio de Janeiro
Programa de Engenharia Elétrica
P.O. Box 68504, Rio de Janeiro, RJ
21.945-970 Brazil
campos@lps.ufrj.br

[2]Escuela Politécnica del Ejército
Facultad de Ingienería Electrónica
Valle de Los Chillos, Quito
Apartado 231-B Ecuador
apolin@espe.edu.ec

ABSTRACT

This paper introduces the constrained version of the Affine Projection Algorithm. The new algorithm is suitable for linearly-constrained minimum-variance applications, which include beamforming and multiuser detection for communications systems. The paper also discusses important aspects of convergence and stability of constrained normalized adaptation algorithms in general. It is shown that normalization may introduce bias in the final solution. Computer simulations are also included providing extra insight to the algorithm performance.

1. INTRODUCTION

Adaptation algorithms which satisfy linear constraints encounter application in several areas of signal processing and communications, such as beamforming, spectral estimation, multiuser detection for communication systems, etc. A robust algorithm which does not require reinitialization and incorporates the constraints into the solution was first introduced by Frost [1]. More recently other constrained adaptation algorithms were introduced which are tailored to specific applications or present advantageous performance regarding convergence and robustness (see, e.g., [2][3]).

The affine-projection (AP) algorithm is among the prominent unconstrained adaptation algorithms that may have a good compromise between fast convergence and low computational complexity. By adjusting the number of projections, or alternatively, the number of reuses, performance can be controlled from that of the normalized least mean squares (NLMS) algorithm to that of the sliding-window recursive least squares (RLS) algorithm [4][5].

In this article we develop and analyze the constrained version of the AP algorithm using the same framework already used for other normalized constrained algorithms, such as the constrained NLMS and BNDR-LMS algorithms. We also show through analysis that normalization may introduce bias.

2. THE AFFINE-PROJECTION ALGORITHM

The Affine Projection (AP) Algorithm updates its coefficient vector such that the new solution belongs to the intersection of L hyperplanes defined by the present and the $L-1$ previous data pairs. The minimization problem used to derive its updating formula is given by

$$\mathbf{w}(k+1) = \min_{\mathbf{w}} \parallel \mathbf{w} - \mathbf{w}(k) \parallel \tag{1}$$

subjected to

$$\mathbf{d}(k) = \mathbf{X}^T(k)\mathbf{w} \tag{2}$$

where

$$\begin{aligned} \mathbf{d}(k) &= [d(k)\; d(k-1)\; \cdots\; d(k-L+1)]^T \\ \mathbf{X}(k) &= [\mathbf{x}(k)\; \mathbf{x}(k-1)\; \cdots\; \mathbf{x}(k-L+1)] \end{aligned} \tag{3}$$

where $\mathbf{x}(k) = [x(k)\; x(k-1)\; \cdots\; x(k-N)]^T$ and $\mathbf{w}$ has order N or $N+1$ elements.

The updating equations for the AP algorithm are obtained from the solution of this minimization problem, and they are presented in Table 1 [6][7]. A step-size (μ) was used to control misadjustment and a small constant (δ) multiplied by the $L \times L$ identity matrix was used to improve robustness.

Table 1: The Affine Projection Algorithm

AP Algorithm
for each k $\{$ $\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k)$ $\mathbf{t}(k) = \left[\mathbf{X}^T(k)\mathbf{X}(k) + \delta\mathbf{I}\right]^{-1}\mathbf{e}(k)$ $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{t}(k)$ $\}$

3. THE CONSTRAINED AFFINE-PROJECTION ALGORITHM

In lineraly constrained adaptive filtering, the constraints are given by the following set of J equations

$$\mathbf{C}^T\mathbf{w}(k) = \mathbf{f} \tag{4}$$

where $\mathbf{C}$ is a $(N+1)\times J$ constraint matrix and $\mathbf{f}$ is a vector containing the J constraint values.

Recalling the constrained LMS algorithm presented by Frost in [1] and realizing that it corresponds to the projection of the unconstrained LMS solution onto the hyperplane defined by (4), we may write

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{P}\mathbf{w}_{LMS}(k+1) + \mathbf{F}\\ &= \mathbf{P}[\mathbf{w}(k) + \mu e(k)\mathbf{x}(k)] + \mathbf{F}\end{aligned}$$

with

$$\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \tag{5}$$

the projection matrix (for a projection onto the homogeneous hyperplane defined by $\mathbf{C}^T\mathbf{w}(k) = \mathbf{0}$) and vector

$$\mathbf{F} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f} \tag{6}$$

used to move the projected solution back to the constraint hyperplane.

The following optimization approach was used to obtain the constrained version of the Affine Projection (CAP) Algorithm

$$\mathbf{w}(k+1) = \min_{\mathbf{w}} \parallel \mathbf{w} - \mathbf{w}(k) \parallel \tag{7}$$

subjected to

$$\begin{cases}\mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w} = \mathbf{0}\\ \mathbf{C}^T\mathbf{w} = \mathbf{f}\end{cases} \tag{8}$$

We can now use Lagrange multipliers in the following objective function to be minimized

$$\begin{aligned}\xi &= [\mathbf{w} - \mathbf{w}(k)]^T [\mathbf{w} - \mathbf{w}(k)]\\ &\quad + \boldsymbol{\lambda}_1^T [\mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}]\\ &\quad + \boldsymbol{\lambda}_2^T [\mathbf{C}^T\mathbf{w} - \mathbf{f}]\end{aligned} \tag{9}$$

and the solution is the equation for the CAP algorithm:

$$\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) + \mathbf{X}(k)\,\mathbf{t}(k)] + \mathbf{F} \tag{10}$$

with

$$\mathbf{t}(k) = [\mathbf{X}^T(k)\,\mathbf{P}\,\mathbf{X}(k)]^{-1}\mathbf{e}(k) \tag{11}$$

and

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k) \tag{12}$$

For $L = 1$ or $L = 2$, the above relations will result in the Constrained NLMS or Constrained BNDR-LMS algorithms [8], respectively. For all constrained algorithms mentioned here, the simplification $\mathbf{P}\mathbf{w}(k) + \mathbf{F} = \mathbf{w}(k)$ should be avoided in a finite precision environment, for accumulation of round-off errors may cause the solution to drift away from the constraint hyperplane [1].

The equations of the Constrained Affine Projection Algorithm are summarized in Table 2, where a step-size $0 < \mu \leq 1$ and a small constant δ were used.

Table 2: The Constrained Affine Projection Algorithm.

CAP Algorithm
for each k
{ $\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k)$
$\mathbf{t}(k) = \left[\mathbf{X}^T(k)\,\mathbf{P}\,\mathbf{X}(k) + \delta\mathbf{I}\right]^{-1}\mathbf{e}(k)$
$\mathbf{w}(k+1) = \mathbf{P}\left[\mathbf{w}(k) + \mu\mathbf{X}(k)\,\mathbf{t}(k)\right] + \mathbf{F}$
}

4. ON THE CONVERGENCE OF THE CAP ALGORITHM

For unconstrained adaptation algorithms, it is usually expected that convergence of the coefficients in the mean can be assured as the number of iterations goes to infinity. For normalized algorithms, such as the NLMS, BNDR-LMS, or quasi-Newton [9] algorithms, convergence with probability one is usually more tractable and is sometimes preferred. As the CAP algorithm is a normalized algorithm, we will favor the latter approach in the analysis to be presented in this section.

Let $\mathbf{w}_o$ be the optimal solution to the constrained optimization problem, i.e.,

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{C}\left(\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\left(\mathbf{C}^T\mathbf{R}^{-1}\mathbf{p} - \mathbf{f}\right) \tag{13}$$

and let $\mathbf{d}(k)$ be modeled as

$$\mathbf{d}(k) = \mathbf{X}^T(k)\mathbf{w}_o \tag{14}$$

If the coefficient-error vector is defined as

$$\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_o \tag{15}$$

then we can easily verify (after using the fact that the optimal solution satisfies the constraints i.e., $\mathbf{f} - \mathbf{C}^T\mathbf{w}_o = \mathbf{0}$), that

$$\begin{aligned}&\Delta\mathbf{w}(k+1) =\\ &\quad \mathbf{P}\{\mathbf{I} - \mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{P}\mathbf{X}(k)]^{-1}\mathbf{X}^T(k)\}\Delta\mathbf{w}(k)\end{aligned} \tag{16}$$

In order to guarantee convergence to zero *everywhere* [10] of the system described by the set of first-order homogeneous equations above, we should ascertain that the transition matrix is time-invariant with all its eigenvalues strictly inside the unit circle. This is clearly not satisfied. Analysis of convergence in the mean requires independence assumptions that are not fulfilled by (16). Furthermore, the projection matrix $\mathbf{P}$ is not a Lyapunov transformation [11] and cannot be used to define a stable equivalent system (in the sense of Lyapunov) as was done in [12]. In fact, if we rewrite (16) as

$$\Delta\mathbf{w}(k+1) = \mathbf{P}\mathbf{T}(k)\Delta\mathbf{w}(k) \tag{17}$$

we notice that the eigenvalues of $\mathbf{T}(k)$ are not necessarily inside the unit circle, although those of $\mathbf{P}\mathbf{T}(k)$ are always either 1 or 0. Therefore, the constant projection matrix $\mathbf{P}$ projects the drifting vector $\mathbf{T}(k)\Delta\mathbf{w}(k)$ onto the subspace orthogonal to the subspace spanned by the constraint matrix $\mathbf{C}$, such that the norm of $\Delta\mathbf{w}(k+1)$ is never greater than the norm of $\Delta\mathbf{w}(k)$. This prevents divergence from the constraint plane, but does not assure that $\|\Delta\mathbf{w}(k)\| \longrightarrow 0$. Unbiasedness is, therefore, not guaranteed.

5. SIMULATIONS RESULTS

A first experiment was carried out in a system identification problem where the filter coefficients were constrained to preserve linear phase at every iteration. For this example we made $N = 10$ and, in order to fulfill the linear phase requirement, we made

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{N/2} \\ \mathbf{0}^T \\ -\mathbf{J}_{N/2} \end{bmatrix} \tag{18}$$

with $\mathbf{J}$ being a reversal matrix (an identity matrix with all lines in reversed order), and

$$\mathbf{f} = [0 \cdots 0]^T \tag{19}$$

This didactic setup was employed to show the improvement of the convergence speed when L is increased. Due to the symmetry of $\mathbf{C}$ and the fact that $\mathbf{f}$ is a null vector, more efficient structures could be used [13]. The input signal consists of zero mean unity variance colored noise with eigenvalue spread around 2000 and the reference signal was obtained after filtering the input by a linear-phase FIR filter and adding an observation noise with variance equal to $1e-10$. Fig. 1 shows the learning curves for the CAP algorithm with values of L varying from 1 to 5. It is also clear from this figure that the misadjustment increases with L.

Also for this first experiment, Fig. 2 shows that we have no bias in the coefficient vector after convergence.

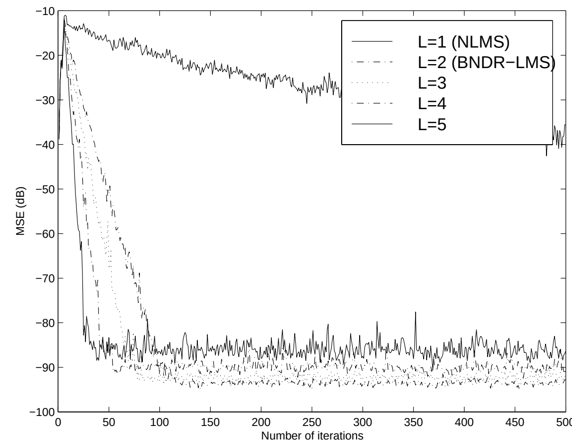

Figure 1: Learning curves for the CAP Algorithm.

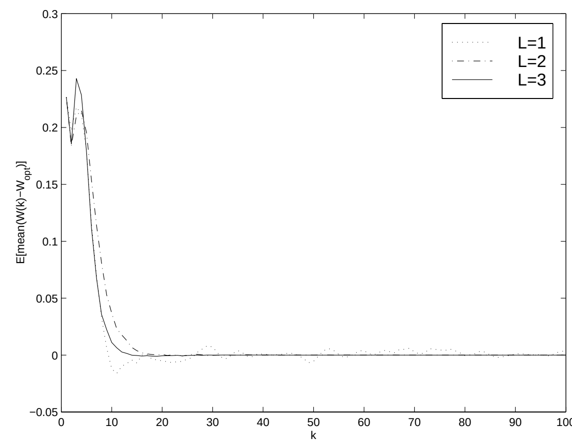

Figure 2: First experiment: no bias in the coefficient vector.

A second experiment was done where the received signal consists of three sinusoids in white noise:

$$x(k) = \sin(0.3k\pi) + \sin(0.325k\pi) + \sin(0.7k\pi) + n(k) \tag{20}$$

where n(k) is white noise with power such that the SNR is $40dB$. The filter is constrained to pass frequency components of 0.1rad/s and 0.25rad/s undistorted which results in the following constraint matrix and vector:

$$\boldsymbol{C}^T = \begin{bmatrix} 1 & \cos(0.2\pi) & \ldots & \cos[(N-1)0.2\pi] \\ 1 & \cos(0.5\pi) & \ldots & \cos[(N-1)0.5\pi] \\ 1 & \sin(0.2\pi) & \ldots & \sin[(N-1)0.2\pi] \\ 1 & \sin(0.5\pi) & \ldots & \sin[(N-1)0.5\pi] \end{bmatrix} \tag{21}$$

$$\boldsymbol{F}^T = [1\ 1\ 0\ 0] \tag{22}$$

The norm of the coefficient-error vector for values of L from 1 to 3 is depicted in Fig. 3. From this figure we can realize that, although faster, the CAP algorithm presents an increasing misadjustment with L, specially when this number of projections is higher than 2 (this value corresponding to the BNDR-LMS algorithm).

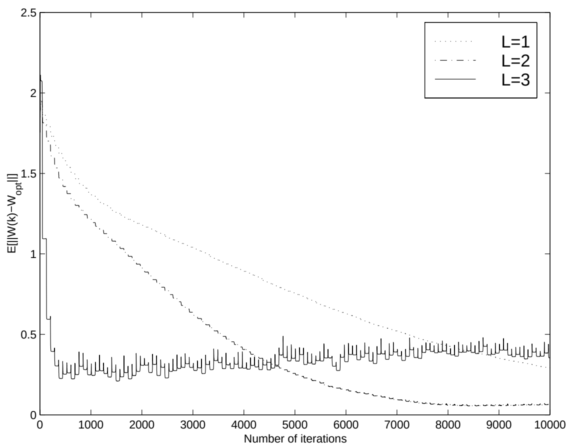

Figure 3: Coefficient-vector deviation for the second experiment.

The reason for this behavior is found in Fig. 4 where are plotted the curves corresponding to the mean value of the coefficient-error vector (averaged in 10 independent trials) for the three first values of L.

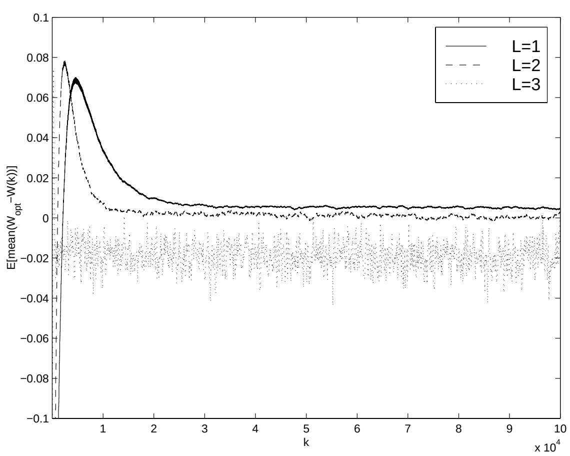

Figure 4: Second experiment: there is an increasing (with L) bias in the coefficient vector.

6. CONCLUSIONS

In this paper, we have introduced the constrained version of the Affine Projection Algorithm. Through analysis, we have verified that this type of algorithm may introduce bias in the coefficient-error vector. The simulation results of two experiments including both cases of biased and unbiased solutions supported the analysis claims and evaluated the performance of the proposed algorithm.

7. REFERENCES

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

[2] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "A fast least-squares algorithm for linearly constrained adaptive filtering," *IEEE Trans. Signal Processing*, vol. 44, pp. 1168–1174, May 1996.

[3] M. L. R. de Campos, S. Werner, and J. A. Apolinário Jr., "Householder-transform constrained LMS algorithms with reduced-rank updating," in *Proc. International Conference of Acoustics, Speech, and Signal Processing*, Phoenix, USA, pp. 1857–1960, 1999.

[4] M. L.R. de Campos, P. S. R. Diniz, and J. A¿ Apolinário Jr., "On normalized data-reusing and affine-projections algorithms," in *Proc. IEEE International Conference on Electronics, Circuits, and Systems*, Pafos, Cyprus, 1999.

[5] M. Montazeri and P. Duhamel, "A set of algorithms linking NLMS and block RLS algorithms," *IEEE Trans. Signal Processing*, vol. 43, pp. 444–453, Feb. 1995.

[6] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan*, vol. 67-A, no. 5, pp. 19–27, 1984.

[7] S. L. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. International Conference of Acoustics, Speech, and Signal Processing*, Detroit, USA, pp. 3023–3026, 1995.

[8] J. A. Apolinário Jr., S. Werner, and P. S. R. Diniz, "Constrained normalized adaptive filters for CDMA mobile communications," in *Proc. European Signal Processing Conference*, vol. IV, Rhodes, Greece, pp. 2053–2056, 1998.

[9] M. L. R. de Campos and A. Antoniou, "Analysis of a new quasi-Newton adaptive filtering algorithm," in *Proc. IEEE International Conference on Electronics, Circuits, and Systems*, Rhodes, Greece, 1996.

[10] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd ed., 1991.

[11] F. R. Gantmacher, *The Theory of Matrices*, vol. II. New York: Chelsea Publishing Company, 1959.

[12] M. L. R. de Campos and A. Antoniou, "A new quasi-Newton adaptive filtering algorithm," *IEEE Trans. Circuits and Systems — Part II*, vol. 44, pp. 924–934, Nov. 1997.

[13] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "Simplified FLS algorithm for linear phase adaptive filtering," in *Proc. European Signal Processing Conference*, vol. III, Rodes, Greece, pp. 1237–1240, 1998.