Low-Complexity Implementation of the Constrained Recursive Least-Squares Algorithm

Reza Arablouei* and Kutluyıl Doğançay†
Institute for Telecommunications Research
†School of Electrical and Information Engineering
University of South Australia, Mawson Lakes 5095 SA, Australia
*E-mail: arary003@mymail.unisa.edu.au Tel: +61-8-830-23316

***Abstract*—A low-complexity implementation of the constrained recursive least squares (CRLS) adaptive filtering algorithm is developed based on the method of weighting and the dichotomous coordinate descent (DCD) iterations. The method of weighting is employed to incorporate the linear constraints into the least squares problem of interest. The DCD iterations are then used to solve the normal equations of the resultant unconstrained least squares problem. The new algorithm has a significantly smaller computational complexity than the CRLS algorithm while delivering convergence performance on par with it. Simulations demonstrate the effectiveness of the proposed algorithm.**

I. INTRODUCTION

Linearly-constrained adaptive filtering algorithms are powerful signal processing tools with widespread use in several applications such as beamforming, blind multiuser detection in code-division multiple-access (CDMA) systems, and system identification [1]. The linear constraints usually emerge from the prior knowledge about the system, e.g., directions of arrival in array processing [2], spreading codes in blind multiuser detection [3], and linear phase in system identification [4]

The constrained least mean squares (CLMS) algorithm [5] is the first linearly-constrained adaptive filtering algorithm. It was originally introduced as an adaptive solution for the well-known problem of linearly-constrained minimum-variance (LCMV) filtering, which is commonly encountered in antenna array processing [2]. Since CLMS belongs to the family of LMS-type stochastic-gradient algorithms, it is simple, robust, and computationally efficient. However, it is also subject to the limitations of this family. Most notably, it has a slow convergence that is further aggravated when the input signal is correlated. The constrained recursive least squares (CRLS) algorithm [6] solves the problem of slow convergence in CLMS at the cost of increased computational complexity. This algorithm is developed by directly incorporating the linear constraints into a recursive least squares solution. The constrained affine projection (CAP) algorithm [7] features complexity and convergence rate between those of the CLMS and CRLS algorithms.

The generalized sidelobe canceler (GSC) [8] is an alternative approach to the implementation of the linearly-constrained adaptive filtering. The GSC offers the advantage of employing any unconstrained adaptive filtering algorithm by transforming the constrained optimization problem to an unconstrained one. However, at each iteration, it multiplies the input vector by a blocking matrix that is orthogonal to the constraint matrix. This generally makes the GSC computationally more demanding than the direct-form constrained adaptive filtering algorithms, e.g., CLMS, CRLS, and CAP, which incorporate the linear constraints into their adaptation process. Householder-transform constrained adaptive filtering [9] is an efficient implementation of the GSC that takes advantage of the effectiveness of the Householder transformation to reduce the complexity. However, it can only be applied to the LCMV filtering where the desired signal is zero. In [10], it is proven that the GSC using the RLS algorithm is mathematically equivalent to the CRLS algorithm. A review and comparison of different RLS-type constrained adaptive filtering algorithms can be found in [12].

In this paper, we develop a new linearly-constrained recursive least squares algorithm employing the method of weighting [11] and the dichotomous coordinate descent (DCD) iterations [13]. The proposed algorithm is a low-complexity implementation of the CRLS algorithm. Despite the approximate nature of the method of weighing and the DCD algorithm, the performance of the new algorithm can be made arbitrary close to that of the CRLS algorithm by appropriate selection of the design parameters. The basic idea behind the new algorithm is to convert the constrained least squares problem to an unconstrained one by applying the method of weighting and to solve the system of linear equations (SLE) of the normal equations corresponding to the unconstrained problem by means of the DCD algorithm. The DCD algorithm is an excellent multiplication-free tool for solving SLEs and falls into the category of shift-and-add algorithms. The number of iterations that the DCD algorithm exercises at each run and the resolution of its step-size control the trade-off between its accuracy and complexity [14].

II. ALGORITHM DESCRIPTION

Let us consider a linear system described by

$$d_n = \boldsymbol{\omega}^T \mathbf{x}_n + v_n$$

This work was supported in part by a *Commonwealth Scientific and Industrial Research Organisation (CSIRO)* scholarship.

where $d_n \in \mathbb{R}$ is the desired signal at time index n, $\boldsymbol{\omega} \in \mathbb{R}^L$ is the vector of unknown system parameters,

$$\mathbf{x}_n = [x_n, x_{n-1}, \dots, x_{n-L+1}]^T,$$

$x_n \in \mathbb{R}$ is the input signal, $v_n \in \mathbb{R}$ represents the background noise, and superscript T stands for matrix transposition. Adaptive filtering algorithms recursively estimate the unknown system parameters exploiting the data available up to the current time instant. Generally, the tap weights of an adaptive filter, denoted by $\mathbf{w}_n \in \mathbb{R}^L$, are taken as the estimate.

In linearly-constrained adaptive filtering, a set of constraints is imposed upon the filter weights at each time instant as

$$\mathbf{C}^T\mathbf{w}_n = \mathbf{f}$$

where $\mathbf{C} \in \mathbb{R}^{L\times K}$ is the constraint matrix ($K \leq L$) and $\mathbf{f} \in \mathbb{R}^K$ is the response vector. A constrained exponentially-weighted least-squares (LS) estimate can be found as

$$\begin{gathered}\mathbf{w}_n = \arg\min_{\boldsymbol{w}} \|\mathbf{d}_n - \mathbf{X}_n^T\boldsymbol{w}\|^2 \\ \text{subject to } \mathbf{C}^T\boldsymbol{w} = \mathbf{f}\end{gathered} \tag{1}$$

where

$$\mathbf{d}_n = \left[d_n, \lambda^{1/2}d_{n-1}, \dots, \lambda^{(n-1)/2}d_1\right]^T,$$

$$\mathbf{X}_n = \left[\mathbf{x}_n, \lambda^{1/2}\,\mathbf{x}_{n-1}, \dots, \lambda^{(n-1)/2}\mathbf{x}_1\right],$$

$\|\cdot\|$ denotes the Euclidean norm, and $0 \ll \lambda \leq 1$ is the forgetting factor. The solution of this optimization problem obtained through the method of Lagrange multipliers is [6]

$$\mathbf{w}_n = \mathbf{R}_n^{-1}\mathbf{p}_n + \mathbf{R}_n^{-1}\mathbf{C}(\mathbf{C}^T\mathbf{R}_n^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^T\mathbf{R}_n^{-1}\mathbf{p}_n) \tag{2}$$

where

$$\mathbf{R}_n = \mathbf{X}_n\mathbf{X}_n^T = \textstyle\sum_{i=1}^n \lambda^{n-i}\mathbf{x}_i\mathbf{x}_i^T = \lambda\mathbf{R}_{n-1} + \mathbf{x}_n\mathbf{x}_n^T$$

is the exponentially-weighted input autocorrelation matrix and

$$\mathbf{p}_n = \mathbf{X}_n\mathbf{d}_n = \textstyle\sum_{i=1}^n \lambda^{n-i}\mathbf{x}_i d_i = \lambda\mathbf{p}_{n-1} + \mathbf{x}_n d_n$$

is the exponentially-weighted cross-correlation vector between the input vector and the desired signal. The constrained recursive least-squares (CRLS) algorithm [6] is a recursive calculation of (2) that avoids the matrix inversions by applying the matrix inversion lemma [15].

The expression of (2) is an exact solution for the constrained LS problem of interest, (1). However, employing the method of weighting [11], an approximate solution, $\widetilde{\mathbf{w}}_n$, can be obtained by solving the following unconstrained optimization problem:

$$\widetilde{\mathbf{w}}_n = \arg\min_{\boldsymbol{w}} \left\| \begin{bmatrix}\mathbf{d}_n \\ \mu\mathbf{f}\end{bmatrix} - \begin{bmatrix}\mathbf{X}_n^T \\ \mu\mathbf{C}^T\end{bmatrix}\boldsymbol{w} \right\|^2$$

where $\mu \gg 1$ is a weighting parameter. Consequently, $\widetilde{\mathbf{w}}_n$ is the solution of the following normal equations [15]:

$$\left([\mathbf{X}_n \quad \mu\mathbf{C}]\begin{bmatrix}\mathbf{X}_n^T \\ \mu\mathbf{C}^T\end{bmatrix}\right)\widetilde{\mathbf{w}}_n = [\mathbf{X}_n \quad \mu\mathbf{C}]\begin{bmatrix}\mathbf{d}_n \\ \mu\mathbf{f}\end{bmatrix}$$

or

$$(\mathbf{R}_n + \mu^2\mathbf{C}\mathbf{C}^T)\widetilde{\mathbf{w}}_n = \mathbf{p}_n + \mu^2\mathbf{C}\mathbf{f},$$

which can be written as

$$\boldsymbol{\Phi}_n\widetilde{\mathbf{w}}_n = \mathbf{z}_n \tag{3}$$

by defining $\boldsymbol{\Phi}_n = \mathbf{R}_n + \mu^2\mathbf{C}\mathbf{C}^T$ and $\mathbf{z}_n = \mathbf{p}_n + \mu^2\mathbf{C}\mathbf{f}$. Using the definition of $\mathbf{p}_n$, it is easy to verify that

$$\mathbf{z}_n = \lambda\mathbf{z}_{n-1} + \mathbf{x}_n d_n + (1-\lambda)\mu^2\mathbf{C}\mathbf{f}. \tag{4}$$

The system of linear equations (SLE) of (3) can be efficiently solved utilizing the dichotomous coordinate descent (DCD) algorithm [13]. However, in order to exploit the full potential of the DCD algorithm, similar to the approach taken in [14], we solve

$$\boldsymbol{\Phi}_n\boldsymbol{\Delta}\widetilde{\mathbf{w}}_n = \boldsymbol{\rho}_n \tag{5}$$

instead of (3) where $\boldsymbol{\Delta}\widetilde{\mathbf{w}}_n = \widetilde{\mathbf{w}}_n - \widehat{\mathbf{w}}_{n-1}$ and

$$\boldsymbol{\rho}_n = \mathbf{z}_n - \boldsymbol{\Phi}_n\widehat{\mathbf{w}}_{n-1} \tag{6}$$

while $\widehat{\mathbf{w}}_{n-1}$ is the vector of filter weights computed at the previous time instant. Substituting (4) into (6) together with using the definitions of $\boldsymbol{\Phi}_n$ and $\mathbf{R}_n$ results in

$$\begin{aligned}\boldsymbol{\rho}_n = \lambda\mathbf{z}_{n-1} + \mathbf{x}_n d_n + (1-\lambda)\mu^2\mathbf{C}\mathbf{f} - \lambda\boldsymbol{\Phi}_{n-1}\widehat{\mathbf{w}}_{n-1} \\ -\mathbf{x}_n\mathbf{x}_n^T\widehat{\mathbf{w}}_{n-1} - (1-\lambda)\mu^2\mathbf{C}\mathbf{C}^T\widehat{\mathbf{w}}_{n-1},\end{aligned}$$

or

$$\begin{aligned}\boldsymbol{\rho}_n = \lambda\mathbf{z}_{n-1} - \lambda\boldsymbol{\Phi}_{n-1}\widehat{\mathbf{w}}_{n-1} + \mathbf{x}_n e_n \\ + (1-\lambda)\mu^2\mathbf{C}(\mathbf{f} - \mathbf{C}^T\widehat{\mathbf{w}}_{n-1})\end{aligned} \tag{7}$$

where the estimation error is defined as

$$e_n = d_n - \widehat{\mathbf{w}}_{n-1}^T\mathbf{x}_n.$$

Since $\widehat{\mathbf{w}}_{n-1}$ is an approximate solution, we can define the residual vector at time instant $n-1$ as

$$\mathbf{r}_{n-1} = \mathbf{z}_{n-1} - \boldsymbol{\Phi}_{n-1}\widehat{\mathbf{w}}_{n-1} \tag{8}$$

Using (8) in (7) yields

$$\boldsymbol{\rho}_n = \lambda\mathbf{r}_{n-1} + \mathbf{x}_n e_n + (1-\lambda)\mu^2\mathbf{C}(\mathbf{f} - \mathbf{C}^T\widehat{\mathbf{w}}_{n-1}).$$

Solving (5) using the DCD algorithm gives both $\boldsymbol{\Delta}\widehat{\mathbf{w}}_n$, an approximate solution for $\boldsymbol{\Delta}\widetilde{\mathbf{w}}_n$, and $\mathbf{r}_n$ at each time instant [14]. Having found $\boldsymbol{\Delta}\widehat{\mathbf{w}}_n$, we can calculate the filter weights as

$$\widehat{\mathbf{w}}_n = \widehat{\mathbf{w}}_{n-1} + \boldsymbol{\Delta}\widehat{\mathbf{w}}_n.$$

We call the proposed algorithm *reduced-complexity constrained recursive least-squares* (RC-CRLS) and summarize it in Table I. We also present the DCD algorithm in Table II where a_n^κ denotes the κth element of a vector $\mathbf{a}_n$ while $\Phi_n^{\kappa,\kappa}$ and $\boldsymbol{\Phi}_n^{(\kappa)}$ are the (κ,κ)th entry and κth column of the matrix $\boldsymbol{\Phi}_n$, respectively.

The accuracy and complexity of the DCD algorithm is controlled by its user-defined parameters, N, M, and H, which establish a trade-off between complexity and performance

[13]. At each run of the algorithm, maximum N iterative updates are performed. In other words, N determines the maximum number of filter coefficients that can be updated at each time instant. Hence, in general, adaptive filtering based on the DCD algorithm implements a form of *selective partial updates* [16]. It is known that by selective partial updating, one can trade performance for complexity [17]. In the DCD algorithm, the step-size α can accept one of M predefined values corresponding to representation of the elements of the solution vector as fixed-point words with M bits within an amplitude range of $[-H, H]$.

III. Computational Complexity

The lower-right $(L-1)\times(L-1)$ block of $\mathbf{R}_n$ can be obtained by copying the upper-left $(L-1)\times(L-1)$ block of $\mathbf{R}_{n-1}$. The only part of the matrix that should be directly updated is the first row and the first column. Due to the symmetry of $\mathbf{R}_n$, it is sufficient to update only the first column via

$$\mathbf{R}_n^{(1)} = \lambda \mathbf{R}_{n-1}^{(1)} + \mathbf{x}_n x_n.$$

Similar to [14], we choose the forgetting factor as $\lambda = 1 - 2^{-s}$ where s is a positive integer. Consequently, multiplications by λ can be replaced by bit-shifts and additions. In addition, since $\mathbf{\Phi}_n$ is symmetric, we only need to compute its upper triangular part.

The CRLS algorithm requires $(3K^2 + 5K + 9)L + K^2 + 2K + 16$ multiplications per iteration while the proposed algorithm requires only $(2K + 3)L$ multiplications per iteration.

IV. Simulations

We consider an LCMV filtering problem. The input signal is made of three sinusoids with random phases, ϕ_1, ϕ_2, and ϕ_3, and normalized frequencies $f_1 = 0.065$, $f_2 = 0.195$, and $f_3 = 0.41$ corrupted by additive white Gaussian noise, η_n, with power of $\sigma_\eta^2 = 10^{-2}$:

$$x_n = \sin(2\pi f_1 n + \phi_1) + \sin(2\pi f_2 n + \phi_2) + \sin(2\pi f_3 n + \phi_3) + \eta_n.$$

We require the filter to have unity response at two frequencies, $f_4 = 0.14$ and $f_5 = 0.325$. Hence, we impose two point constraints and two first-order derivative constraints [1] apropos f_4 and f_5 and form the constraint matrix as

$$\mathbf{C} = \begin{bmatrix} 1, \cos(2\pi f_4), \ldots, \cos\{2\pi f_4(L-1)\} \\ 1, \cos(2\pi f_5), \ldots, \cos\{2\pi f_5(L-1)\} \\ 0, \sin(2\pi f_4), \ldots, \sin\{2\pi f_4(L-1)\} \\ 0, \sin(2\pi f_5), \ldots, \sin\{2\pi f_5(L-1)\} \end{bmatrix}$$

and the response vector as $\mathbf{f} = [1,1,0,0]^T$. The filters are assumed to be of length $L = 9$. The results are obtained by ensemble-averaging over 10^3 independent runs. The desired signal is zero. The optimal solution is calculated as $\boldsymbol{\omega} = \mathbf{\Psi}^{-1}\mathbf{C}(\mathbf{C}^T\mathbf{\Psi}^{-1}\mathbf{C})^{-1}\mathbf{f}$ where $\mathbf{\Psi} = E[\mathbf{x}_n \mathbf{x}_n^T]$ and $E[\cdot]$ denotes the expectation operation [4]. We also set $\lambda = 0.99$, $M = 16$, and $H = 0.1$.

In Fig. 1, we compare the misalignment curves, i.e., $E[\|\boldsymbol{\omega} - \mathbf{w}_n\|^2]$, of the CRLS algorithm and the proposed algorithm with different values of N and $\mu = 30$. In Fig. 2, we plot the constraint mismatch, i.e., $E[\|\mathbf{C}^T\mathbf{w}_n - \mathbf{f}\|^2]$, of the proposed algorithm as a function of μ with $N = 16$. Fig. 3 depicts the frequency response of the filters after 400 iterations when $\mu = 20$ and only a single iteration of the DCD algorithm is exercised at each time instant, i.e., $N = 1$.

In this experiment, the numbers of required multiplications and additions at each iteration by the CRLS algorithm are respectively 733 and 635 while for the proposed algorithm with $N = 16$ the respective numbers are 99 and 491.

V. Conclusion

We have used the method of weighting and the DCD iterations to develop a new constrained recursive least squares algorithm, which we call RC-CRLS. The new algorithm requires considerably less arithmetic operations than the previously proposed CRLS algorithm while, despite the approximate nature of the method of weighting and the DCD algorithm, it can be easily tuned to perform almost as well as CRLS. This can be explained in view of the fact that, at each iteration, an LS-type adaptive filter seeks an approximate solution to an overdetermined problem based on current and previous noisy observations. Hence, if the imprecision introduced by an incorporated approximate technique, such as the method of weighting, constitutes only a small portion of the total error, it will not have a noticeable impact on the overall performance of the adaptive filter.

We observe that a larger value of the weighting parameter, μ, results in a lower constraint mismatch, i.e., better satisfaction of the constraints. However, it should be noted that a large μ can make the associated SLE hard to solve necessitating a large N, hence requiring a large number of addition operations at each iteration. Therefore, there exists a trade-off between the convergence performance, satisfaction of the constraints, and complexity of the proposed algorithm.

References

[1] M. L. R. de Campos, S. Werner, and J. A. Apolinário Jr., "Constrained adaptive filters," in *Adaptive Antenna Arrays: Trends and Applications*, S. Chrandran, Ed. New York: Springer-Verlag, 2004.

[2] H. L. Van Trees, *Detection, Estimation, and Modulation Theory Part IV: Optimum Array Processing*, New York: Wiley, 2002.

[3] S. Verdu, *Multiuser Detection*, Cambridge, U.K.: Cambridge University Press, 1998.

[4] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*, 3rd ed., Boston: Springer, 2008.

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

[6] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "A fast least-squares algorithm for linearly constrained adaptive filtering," *IEEE Trans. Signal Process.*, vol. 44, no. 5, pp. 1168–1174, May 1996.

[7] S. Werner, J. A. Apolinário Jr., M. L. R. de Campos, and P. S. R. Diniz, "Low-complexity constrained affine-projection algorithms," *IEEE Trans. Signal Process.*, vol. 53, no. 12, pp. 4545–4555, Dec. 2005.

[8] L. J. Griffiths and C. W. Jim, “An alternative approach to linearly constrained adaptive beamforming,” *IEEE Trans. Antennas and Propag.*, vol. AP-30, no. 1, pp. 27–34, Jan. 1982.

[9] M. L. R. de Campos, S. Werner, and J. A. Apolinário Jr., “Constrained adaptation algorithms employing householder transformation,” *IEEE Trans. Signal Process.*, vol. 50, no. 9, pp. 2187–2195, Sep. 2002.

[10] S. Werner, J. A. Apolinário Jr., and M. L. R. de Campos, “On the equivalence of RLS implementations of LCMV and GSC processors,” *IEEE Signal Process. Lett.*, vol. 10, no. 12, pp. 356–359, Dec. 2003.

[11] C. Van Loan, “On the method of weighting for equality-constrained least-squares problems,” *SIAM J. Numer. Anal.*, vol. 22, pp. 851–864, Oct. 1985.

[12] A. A. L. Ramos, J. A. Apolinário Jr., and M. L. R. Campos “On numerical robustness of constrained RLS-like algorithms,” in *Proc. XXI Simpósio Brasileiro de Telecomunicac□ões*, Belém, Brazil, Sep. 2004.

[13] Y. V. Zakharov and T. C. Tozer, “Multiplication-free iterative algorithm for LS problem,” *Electron. Lett.*, vol. 40, no. 9, pp. 567–569, Apr. 2004.

[14] Y. Zakharov, G. White, and J. Liu, “Low complexity RLS algorithms using dichotomous coordinate descent iterations,” *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3150–3161, Jul. 2008.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, third ed., Baltimore, MD: Johns Hopkins University Press, 1996.

[16] K. Doğançay and O. Tanrıkulu, “Adaptive filtering algorithms with selective partial updates,” *IEEE Trans. Circuits Syst. II*, vol. 48, no. 8, pp. 762–769, Aug. 2001.

[17] K. Doğançay, *Partial-Update Adaptive Signal Processing: Design, Analysis and Implementation*, Oxford, UK: Academic Press, 2008.

TABLE I
THE RC-CRLS ALGORITHM.

Initialization

$\mathbf{R}_0 = \delta \mathbf{I}_L$

$\hat{\mathbf{w}}_0 = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$

$\mathbf{r}_0 = \mathbf{0}$

$\mathbf{T} = \mu^2 \mathbf{C}\mathbf{C}^T$

$\mathbf{U} = (1-\lambda)\mu^2 \mathbf{C}$

At iteration $n = 1,2,\ldots$

$\mathbf{R}_n = \lambda \mathbf{R}_{n-1} + \mathbf{x}_n \mathbf{x}_n^T$

$\mathbf{\Phi}_n = \mathbf{R}_n + \mathbf{T}$

$e_n = d_n - \hat{\mathbf{w}}_{n-1}^T \mathbf{x}_n$

$\boldsymbol{\rho}_n = \lambda \mathbf{r}_{n-1} + \mathbf{x}_n e_n + \mathbf{U}(\mathbf{f} - \mathbf{C}^T \hat{\mathbf{w}}_{n-1})$

solve $\mathbf{\Phi}_n \Delta\hat{\mathbf{w}}_n = \boldsymbol{\rho}_n$ and obtain $\Delta\hat{\mathbf{w}}_n$ and $\mathbf{r}_n$

$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \Delta\hat{\mathbf{w}}_n$

TABLE II
THE DCD ALGORITHM SOLVING $\mathbf{\Phi}_n \Delta\tilde{\mathbf{w}}_n = \boldsymbol{\rho}_n$.

Initialization

$m = 1, \alpha = H/2$

$\Delta\hat{\mathbf{w}}_n = \mathbf{0}$

$\mathbf{r}_n = \boldsymbol{\rho}_n$

For $k = 1,2,\ldots,N$

$p = \arg\max_{i=1,\ldots,L}\{|r_n^i|\}$

while $|r_n^p| \le \frac{\alpha}{2}\Phi_n^{p,p}$ and $m \le M$

$m = m+1, \alpha = \alpha/2$

if $m > M$

algorithm stops

$\Delta\hat{w}_n^p = \Delta\hat{w}_n^p + \mathrm{sign}\{r_n^p\}\alpha$

$\mathbf{r}_n = \mathbf{r}_n - \mathrm{sign}\{r_n^p\}\alpha \mathbf{\Phi}_n^{(p)}$

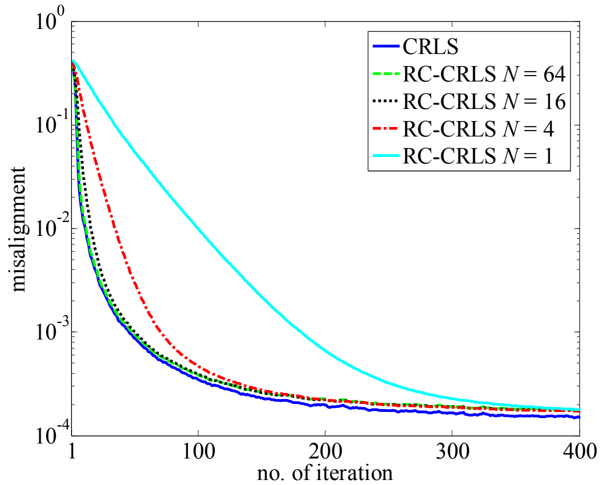

Fig. 1. Misalignment of the CRLS algorithm and the RC-CRLS algorithm with different values of N in an LCMV filtering experiment.

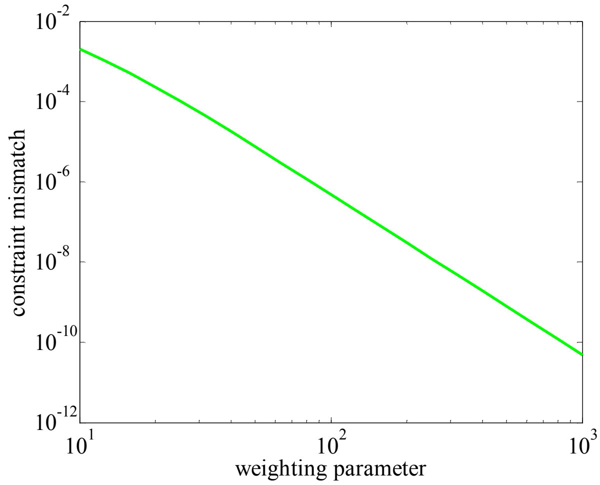

Fig. 2. Constraint mismatch of the RC-CRLS algorithm as a function of μ in an LCMV experiment.

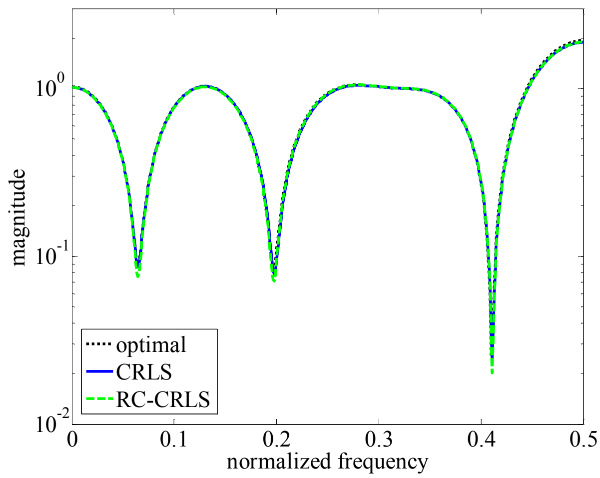

Fig. 3. Frequency response of the filters after 400 iterations.