# Constrained Adaptive Estimation Using Interior Point Optimization

Kaywan H. Afkhamie†, Tim Davidson†, Zhi-Quan Luo† and K. Max Wong†

† Dept. of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4K1

***Abstract*— Linearly constrained adaptive filtering problems occur in applications such as adaptive digital beamforming. The main drawbacks with existing algorithms are slow convergence in the case of LCMV (Frost, 1972) and numerical stability (RLS based algorithms). In this paper we present a novel approach to the problem using techniques from interior point optimization. The proposed algorithm uses a logarithmic barrier function and a single exact Newton step to compute the new parameter vector. Its convergence speed matches that of the RLS based methods but without the associated problems of numerical instability.**

## 1 Introduction

In many adaptive filtering problems the estimated parameters must also satisfy some linear constraints. For example, in *adaptive digital beamforming*, the beamformer can be modeled as an adaptive transversal filter [1, 2], and the goal is to minimize the energy of the interfering signals, and to protect the target signal (steering capability). This requirement for steering capability gives rise to an adaptive filtering problem with some linear constraints.

To formalize the problem, let $\{x(\cdot)\}$ be a sequence of input symbols and $\{d(\cdot)\}$ a desired response sequence. The objective is to identify a linear filter of length $M$, such that the filter's response to the input sequence $\{x(\cdot)\}$ is in some sense close to the desired response, subject to some given linear constraints. When the exponentially weighted mean square error criterion is used, the problem becomes

$$\begin{aligned} \min \quad & \mathcal{F}_n := \frac{1}{n}\sum_{i=1}^{n} \lambda^{n-i} |d(i) - \mathbf{x}_i^T \mathbf{h}|^2, \\ \text{s.t.} \quad & \mathbf{C}^T \mathbf{h} = \mathbf{f}, \ \mathbf{h} \in \Re^M, \end{aligned} \tag{1.1}$$

where $\lambda$ is the forgetting factor, $\mathbf{x}_i^T = [x(i), x(i-1), ..., x(i+1-M)]$ and $\mathbf{h}$ is the vector of tap weights of the linear FIR filter. The matrix $\mathbf{C}$ and vector $\mathbf{f}$ are of appropriate dimensions and they represent the linear (side) constraints for the adaptive filter.

The most widely used technique for solving (1.1) is the "Linearly Constrained Minimum Variance" (LCMV) algorithm by Frost [1]. This is essentially a constrained version of the LMS algorithm, inheriting LMS's simplicity and computational efficiency, but also that algorithms convergence problems in the case of correlated input signals. Other methods have been developed recently that make use of the adaptation gain (or Kalman Gain) readily computed by the RLS algorithm. In fact, [3] suggests that the adaptation gain can be updated using a fast least squares (FLS) algorithm (*e.g.*, [4]). Then, fast convergence is obtained while maintaining a computational complexity of $O(M)$ (RLS is $O(M^2)$). The drawback of RLS based methods is that numerical stability is not guaranteed.

In this paper, we present a novel technique for the linearly constrained adaptive estimation problem utilizing recently developed interior point optimization methods. The proposed algorithm uses a logarithmic barrier function and a single exact Newton step to compute the new parameter vector. Its convergence speed matches that of the RLS based method and there seem to be no problems with numerical instability. The numerical stability is maintained even for relatively low forgetting factors, thus one can trade steady state performance for adaptation speed by changing the forgetting factor.

The paper is organized as follows: in Section 2 we summarize some existing approaches to the constrained adaptive filtering problem. Our proposed algorithm is described in Section 3 and in the following section we present some numerical simulations to illustrate the potential of our methods.

## 2 Previous Methods for Linearly Constrained Adaptive Filtering

We briefly review the previous methods to facilitate the comparison of Section 4. Consider first the LCMV algorithm of Frost [1]. This method is a constrained version of the well known LMS algorithm. The weight vector is updated according to

$$\mathbf{h}_{n+1} = \mathbf{P}[\mathbf{h}_n - \mu\hat{d}(n)\mathbf{x}_n^T] + \mathbf{F}. \qquad (2.2)$$

$\mathbf{P} = \mathbf{I}_M - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T$ and $\mathbf{F} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$ are used to project the unconstrained estimate onto the space of vectors that do satisfy the constraints.

Next we consider an RLS based estimator. Solving the constrained minimization (1.1) by the method of Lagrange multipliers leads to the following solution,

$$\mathbf{h}_n = \mathbf{R}_{xx}^{-1}(n)\mathbf{p}_{xd}(n) + \mathbf{R}_{xx}^{-1}(n)\mathbf{C}[\mathbf{C}^T\mathbf{R}_{xx}^{-1}(n)\mathbf{C}]^{-1} (\mathbf{f} - \mathbf{C}^T\mathbf{R}_{xx}^{-1}(n)\mathbf{p}_{xd}(n)) \qquad (2.3)$$

$\mathbf{R}_{xx}(n)$ and $\mathbf{p}_{xd}(n)$ are the input signal autocorrelation matrix and the cross-correlation vector of desired and input signal, respectively (both are exponentially weighted).

$\mathbf{R}_{xx}^{-1}(n)\mathbf{p}_{xd}(n)$ can be identified as the solution of the unconstrained problem and the second term in (2.3) is a correction in order to satisfy the constraints. Using the definitions

$$\begin{aligned} \mathbf{h}_n^{unc} &= \mathbf{R}_{xx}^{-1}(n)\mathbf{p}_{xd}(n) \\ \Gamma_n &= \mathbf{R}_{xx}^{-1}(n)\mathbf{C} \\ \Psi_n &= \mathbf{C}^T\Gamma_n \end{aligned}$$

we rewrite the Lagrange solution as

$$\mathbf{h}_n = \mathbf{h}_n^{unc} + \Gamma_n\Psi_n^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{h}_n^{unc}] \qquad (2.4)$$

It is well known that the adaptation gain, $\mathbf{g}_n$, computed in the RLS algorithm can be used to update $\mathbf{h}_n^{unc}$,

$$\mathbf{h}_{n+1}^{unc} = \mathbf{h}_n^{unc} + \mathbf{g}_{n+1}[d_{n+1} - \mathbf{h}_n^{unc}\mathbf{x}_{n+1}]. \quad (2.5)$$

Note that in RLS $\mathbf{g}_n$ is used in the recursive update of $\mathbf{R}_{xx}^{-1}(n)$. Since $\Gamma_n$ and $\Psi_n^{-1}$ are closely related to $\mathbf{R}_{xx}^{-1}(n)$ we can very easily derive formulae for their recursive update as well [3]

$$\begin{cases} \Gamma_{n+1} = \dfrac{1}{\lambda}[\Gamma_n - \mathbf{g}_{n+1}\mathbf{x}_{n+1}^T\Gamma_n], \\ \Psi_{n+1}^{-1} = \dfrac{1}{\lambda}[\Psi_n^{-1} + \mathbf{l}_{n+1}\mathbf{x}_{n+1}^T\Gamma_n\Psi_n^{-1}], \end{cases} \qquad (2.6)$$

where

$$\mathbf{l}_{n+1} = \frac{1}{\lambda}[\mathbf{C}^T\Gamma_{n+1}]^{-1}\mathbf{C}^T\mathbf{g}_{n+1}. \qquad (2.7)$$

Equations (2.4)–(2.7) already represent an algorithm for solving (1.1). Notice, however, that the constraint vector $\mathbf{f}$ never enters the algorithm except at the initialization (not shown here), thus the algorithm would soon diverge due to round-off errors. A mechanism to enforce the linear constraints is given in [3].

A new term (just like in (2.4)) is added to correct for numerical errors, this time using the explicit error in the constraints $[\mathbf{f} - \mathbf{C}^T\mathbf{h}_n]$ as the direction for correction. Suppose $\mathbf{h}_{bc}$ is the parameter vector *before correction.* Then the final estimate is found as

$$\mathbf{h}_{n+1} = \mathbf{h}_{bc} + \Gamma_{n+1}\Psi_n^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{h}_n]. \qquad (2.8)$$

We shall refer to this algorithm as LCRLS (linearly constrained recursive least squares).

## 3 Interior Point Column Generation Methods for Adaptive Filtering

Interior Point Column Generation (IPCG) algorithms have been developed in recent years by the optimization community for the problem of finding a feasible point in a set defined by a (possibly infinite) number of convex inequalities (convex feasibility problems) [5]. These algorithms are also known as interior point cutting plane methods when the cuts are linear. At each iteration they compute an approximate center of a current set defined by the inequalities generated in previous iterations. If this approximate center is not a feasible point, then a new inequality (or cutting plane) is placed through the current set. As the number of cuts increases, the set defined by their intersection shrinks and the algorithm gets closer to finding a point satisfying all the convex feasibility constraints.

IPCG algorithms are well suited to adaptive filtering for several reasons. The error minimization criterion (1.1) used in the least squares formulation of the adaptive filtering problem can be easily transformed into a quadratically constrained convex feasibility problem. In particular, we can apply IPCG to find a filter $\mathbf{h}$, such that for all $n \geq 1$,

$$\left\{ \begin{array}{rcl} \frac{1}{n}\sum_{i=1}^{n}\lambda^{n-i}|d(i)-\mathbf{x}_i^T\mathbf{h}|^2 & \leq & \tau_n^2, \\ \|\mathbf{C}^T\mathbf{h}-\mathbf{f}\|^2 & \leq & \epsilon^2, \\ \|\mathbf{h}\|^2 & \leq & R^2, \end{array} \right. \quad (3.9)$$

where the threshold $\tau_n^2$ can be either a constant related to the average noise power or determined adaptively, $\epsilon^2$ is a specified tolerance within which constraints must be satisfied, and $R$ is a normalizing constant to ensure $\mathbf{h}$ is bounded. Thus, the adaptive filtering problem can be formulated as the problem of finding a feasible point satisfying the convex quadratic inequalities given by (3.9). Notice that the first inequality in (3.9) is generated dynamically as new data arrives in a sequential fashion. We shall maintain exactly three convex inequalities in the feasible region $\Omega_n$ (defined below) throughout the computation. This property is very important in the computational aspects of the algorithm as will become clear shortly.

We now propose two new interior point methods for the linearly constrained adaptive filtering problem (1.1). These methods are based on ideas from the IPCG algorithm.

### Algorithm IPM1

According to our discussion above we formulate a convex feasibility problem as follows. At each instant $n$ we look for a filter $\mathbf{h}_n$ at the center of

$$\Omega_n = \{\mathbf{h} \in \Re^M \mid \mathcal{F}_n(\mathbf{h}) \leq \tau_n^2, \\ \|\mathbf{C}^T\mathbf{h}-\mathbf{f}\|^2 \leq \epsilon^2, \|\mathbf{h}\|^2 \leq R^2\}.$$

The threshold $\tau_n^2$ is updated in such a way that the filter $\mathbf{h}_{n-1}$ remains in the interior of the search region $\Omega_n$. But since $\mathbf{h}_{n-1}$ might be close to the boundary we need to re-center it by performing a Newton step on the *logarithmic barrier function* of $\Omega_n$. The latter is defined as

$$\begin{aligned} \phi_{n,\tau}(\mathbf{h}) &= -\log(\tau_n^2 - \mathcal{F}_n(\mathbf{h})) \\ &\quad -\log(\epsilon^2 - \|\mathbf{C}^T\mathbf{h}-\mathbf{f}\|^2) \\ &\quad -\log(R^2 - \|h\|^2). \end{aligned}$$

The function $\phi_{n,\tau}(\mathbf{h})$ is convex and approaches infinity on the boundary of $\Omega_n$. The (unique) global minimizer of $\phi_{n,\tau}(\mathbf{h})$ is called the *analytic center* of $\Omega_n$. The gradient and Hessian of $\phi_{n,\tau}(\mathbf{h})$ are given by

$$\nabla\phi_{n,\tau}(\mathbf{h}) = \frac{\nabla\mathcal{F}_n}{s_1} + \frac{2\mathbf{C}(\mathbf{C}^T\mathbf{h}-\mathbf{f})}{s_2} + \frac{2\mathbf{h}}{s_3},$$

$$\begin{aligned} \nabla^2\phi_{n,\tau}(\mathbf{h}) &= \frac{(\nabla\mathcal{F}_n)^T\nabla\mathcal{F}_n}{s_1^2} + \frac{\nabla^2\mathcal{F}_n}{s_1} \\ &\quad + \frac{4\mathbf{C}(\mathbf{C}^T\mathbf{h}-\mathbf{f})(\mathbf{C}^T\mathbf{h}-\mathbf{f})^T\mathbf{C}^T}{s_2^2} + \frac{2\mathbf{C}\mathbf{C}^T}{s_2} \\ &\quad + \frac{4\mathbf{h}^T\mathbf{h}}{s_3^2} + \frac{2\mathbf{I}}{s_3} \end{aligned}$$

where $s_1 := \tau_n^2 - \mathcal{F}_n(\mathbf{h})$, $s_2 := \epsilon^2 - \|\mathbf{C}^T\mathbf{h}-\mathbf{f}\|^2$, and $s_3 := R^2 - \|\mathbf{h}\|^2$. The gradient and Hessian of $\mathcal{F}_n(\mathbf{h})$ are given by

$$\begin{aligned} \nabla\mathcal{F}_n &= -\mathbf{p}_{xd}(n) + \mathbf{R}_{xx}(n)\mathbf{h}, \\ \nabla^2\mathcal{F}_n &= \mathbf{R}_{xx}(n). \end{aligned}$$

The steps of IPM1 are given as follows:

**Step 1.** Initialization: let $\epsilon$, $\beta > 0$, $\tau_0$, $R$, $\mathbf{C}$, $\mathbf{f}$ and $\mathbf{h}_0 \in \Omega_0$ be given.
**Step 2.** Updating: let $\mathbf{h}_{n-1}$ be the filter we have from the previous iteration. Update $\tau_n$ as follows:

$$\tau^2 := \mathcal{F}_n(\mathbf{h}_{n-1}) + \beta r_n, \quad \text{where}$$
$$r_n = \sqrt{[\nabla\mathcal{F}_n(\mathbf{h}_{n-1})]^T[\nabla^2\phi_{0,\tau}]^{-1}[\nabla\mathcal{F}_n(\mathbf{h}_{n-1})]} \quad (3.10)$$

With this choice of $\tau_n$ we ensure that $s_1 = \tau_n^2 - \mathcal{F}_n(\mathbf{h}_{n-1})$ is a positive number.
**Step 3.** Centering: For $n > 0$, perform Newton iterations starting from $\mathbf{h}_{n-1}$:

$$\mathbf{h} := \mathbf{h} - (\nabla^2\phi_{n,\tau})^{-1}(\mathbf{h})\nabla\phi_{n,\tau}(\mathbf{h})$$

until $\phi_{n,\tau}$ is (approximately) minimized. Repeat Step 2. □

It can be shown theoretically that only a constant number of Newton iterations are needed to approximately minimize the the logarithmic barrier function $\phi_{n,\tau}(\mathbf{h})$. If constant $\beta$ is chosen large enough, only one Newton iteration is needed. In practice, we set $\beta = 1$ and perform only one Newton iteration. This leads to substantial computational saving since it makes recursive updating of $s_1$ possible. Indeed, the updating of $\tau_n^2$ implies the new $s_1$ is given by $\beta r_n$, and the quantities $\nabla \mathcal{F}_n$, $\nabla^2 \mathcal{F}_n$ can be updated recursively (in the usual manner). This makes the **IPM1** a recursive algorithm in the sense its complexity per iteration is $O(M^3)$, independent of $n$.

### Algorithm IPM2

Given a positive scalar $\mu > 0$, let us define a weighted logarithmic barrier function for $\Omega_n$:

$$\psi_{n,\mu}(\mathbf{h}) := -\log(\tau^2 - \mathcal{F}_n(\mathbf{h})) \\ -\mu \log(\epsilon^2 - \|\mathbf{C}^T\mathbf{h} - \mathbf{f}\|^2) - \mu \log(R^2 - \|\mathbf{h}\|^2).$$

The function $\psi_{n,\mu}$ is convex over $\Omega_n$ and has a unique minimum which we call $\mu$-analytic center for $\Omega_n$. It can be shown that for fixed $n$, the $\mu$-analytic center approaches the minimizer of adaptive filtering problem (1.1), as $\mu \to 0^+$.

The gradient and the Hessian of $\psi_{n,\mu}$ are given by

$$\nabla \psi_{n,\mu}(\mathbf{h}) = \frac{\nabla \mathcal{F}_n}{s_1} + \frac{2\mu \mathbf{C}(\mathbf{C}^T\mathbf{h} - \mathbf{f})}{s_2} + \frac{2\mu \mathbf{h}}{s_3}$$

$$\nabla^2 \psi_{n,\mu}(\mathbf{h}) = +\frac{(\nabla \mathcal{F}_n)^T \nabla \mathcal{F}_n}{s_1^2} + \frac{\nabla^2 \mathcal{F}_n}{s_1} \\ +\mu \frac{4\mathbf{C}(\mathbf{C}^T\mathbf{h} - \mathbf{f})(\mathbf{C}^T\mathbf{h} - \mathbf{f})^T \mathbf{C}^T}{s_2^2} + \frac{2\mu \mathbf{C}\mathbf{C}^T}{s_2} \\ +\mu \frac{4\mathbf{h}^T\mathbf{h} + 2 s_3 \mathbf{I}}{s_3^2}.$$

Unlike IPM1 which adjusts $\tau$ at each iteration, IPM2 adjusts the weight $\mu$ to drive it to zero. The threshold value $\tau$ is kept constant. The details of the algorithmic steps are summarized below:

**Step 1.** Initialization: let $\epsilon$, $\tau$, $\beta$, $R$, $\mathbf{C}$, $\mathbf{f}$ and $\mathbf{h}_0 \in \Omega_0$ be given. Let $\mu = 1$.
**Step 2.** Centering: For $n \geq 0$, perform Newton iterations starting from $\mathbf{h}_n$:

$$\mathbf{h} := \mathbf{h} - (\nabla^2 \psi_{n,\mu})^{-1}(\mathbf{h}) \nabla \psi_{n,\mu}(\mathbf{h})$$

untill $\psi_{n,\mu}$ is (approximately) minimized.
**Step 3.** Updating: let $\mathbf{h}_{n+1}$ be the output of Step 2. Update the cost function $\mathcal{F}(\mathbf{h})$ using the newly received channel data. Update $\mu$ as follows:

$$\mu := \beta \mu \tag{3.11}$$

Repeat Step 2. □

Similar to the case of IPM1, it can be shown theoretically that only a constant number of Newton iterations are needed to approximately minimize the the logarithmic barrier function $\phi_{n,\tau}(\mathbf{h})$ when $\mu$ is updated by (3.11).

## 4 Simulation

The experiment we use is the same as the one presented in [3]. The input signal is composed of three sinusoids at normalized frequencies 0.15, 0.1625, and 0.35, and white noise is added at an SNR of 40 dB,

$$x(n) = \sin(0.3n\pi) + \sin(0.325n\pi) \\ + \sin(0.7n\pi) + b(n).$$

The filter has length $M = 11$ and is constrained to have unity response at frequencies 0.1 and 0.25. This produces the constraint parameters

$$\mathbf{C}^T = \begin{bmatrix} 1 & \cos(0.2\pi) & \dots & \cos((M-1)0.2\pi) \\ 1 & \cos(0.5\pi) & \dots & \cos((M-1)0.5\pi) \\ 0 & \sin(0.2\pi) & \dots & \sin((M-1)0.2\pi) \\ 0 & \sin(0.5\pi) & \dots & \sin((M-1)0.5\pi) \end{bmatrix}$$

and

$$\mathbf{f} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T.$$

We tested our proposed algorithm IPM1 against Frost's LCMV and the LCRLS algorithm of Resende *et al.* (IPM2 performs similarly to IPM1 by proper choice of parameters $\mu$ and $\beta$). We first investigate the stability issue. Figure 1 shows the convergence of the estimated filter sequence to the ideal constrained optimal solution. Clearly, both the LCRLS and IPM1 converge much quicker than LCMV. But after about 3000 iterations the LCRLS algorithm becomes unstable, whereas the new method shows no sign of instability. The forgetting factor here is $\lambda = 0.99$, and LCMV uses a step size of $\mu = 0.1$ for "fast" convergence. The result is averaged over 100 Monte Carlo trials. Figure 2 shows the frequency responses of the filters after 4000 iterations. Other papers have also observed the instability of LCRLS. As [6] points out, the instability is in fact due to the correction mechanism in (2.8) and not caused by explosive divergence of the RLS algorithm.

We also tested the proposed algorithm in a more difficult scenario, as far as stability is concerned. Reducing the forgetting factor to 0.7 we ran an input sequence of length 200,000. While the estimated filter does not separate the two closely spaced frequencies anymore the filter remains stable and matches the constraints throughout the length of the simulation (see Figure 3).

We have shown the stability of IPM1, regardless of the forgetting factor $\lambda$. This property can be further exploited in applications where the system can undergo abrupt changes. With the IPM1 method we may trade in some estimation precision for faster adaptation without fear of instability. Figure 4 shows the convergence of all three algorithms after an abrupt change occurs at iteration 1000. Before $n = 1000$ the signal frequencies are 0.325, 0.350, 0.375 and then they switch to the familiar values of previous experiments. It was necessary in this case to reduce the step size $\mu$ in the LCMV to 0.03. At $\mu = 0.1$, LCMV would diverge immediately. The forgetting factors are 0.995 and 0.95 for LCRLS and IPM1, respectively. This allows IPM1 to adjust to the new signal within about 200 samples, while LCRLS takes more than 1500 samples to get to the same value. Although, the forgetting factor has been increased to 0.995, LCRLS is still not stable – the divergence occurs just after iteration 6000 (not shown in figures).

A note about computational complexity: the current implementation of the IPM algorithm has about twice the complexity of the LCRLS algorithm (implemented with RLS for computing the adaptation gain). Both algorithms are significantly more expensive to compute than Frost's LCMV algorithm.

## 5 Conclusions

We have proposed an alternative to the classical LCMV algorithm for estimating an adaptive filter under linear constraints. A comparison with the work of Resende *et al.* shows that our method matches the convergence speed of RLS based methods, however, the Interior Point method leads to an inherently stable solution of the constrained filtering problem. The method is flexible as it can handle arbitrary numbers of either linear or convex quadratic constraints.

There are other RLS based algorithms that may be numerically stable. Leung [6] mentions a QRD-RLS algorithm by Shepherd and McWhirter. In further work it would be interesting to investigate in more detail the stability, numerical complexity and convergence properties of IPM as compared to QRD-RLS. Another issue is robustness to initialization. It is now understood that RLS is quite sensitive to its initialization [7]; IPM however, can be initialized almost arbitrarily at least in the unconstrained case.

## References


[1] O. L. Frost, "An Algorithm for Linearly Constrained Adaptive Array Processing," *Proceedings of the IEEE*, vol. 60, pp. 926–935, Aug. 1972.

[2] S. Haykin, *Adaptive Filter Theory.* NJ: Prentice Hall, third ed., 1996.

[3] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "A Fast Least-Squares Algorithm for Linearly Constrained Adaptive Filtering," *IEEE Transactions on Signal Processing*, vol. 44, pp. 1168–1174, May 1996.

[4] D. T. M. Slock and T. Kailath, "Numerically stable fast transversal filters for recursive least-squares adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 38, pp. 92–114, Jan. 1991.

[5] Z.-Q. Luo and J. Sun, "An Analytic Center based Column Generation Algorithm for Convex Quadratic Feasibility Problem," *accepted for publication in SIAM J. Optimization*, 1997.

[6] J. Huo and Y. H. Leung, "Numerical Properties of the linearly constrained QRD-RLS Adaptive Filter," in *Proceedings of ICASSP 1998*, (Seattle, WA), May 1998.

[7] G. V. Moustakides, "Study of the transient phase of the forgetting factor RLS," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2468–2476, October 1997.

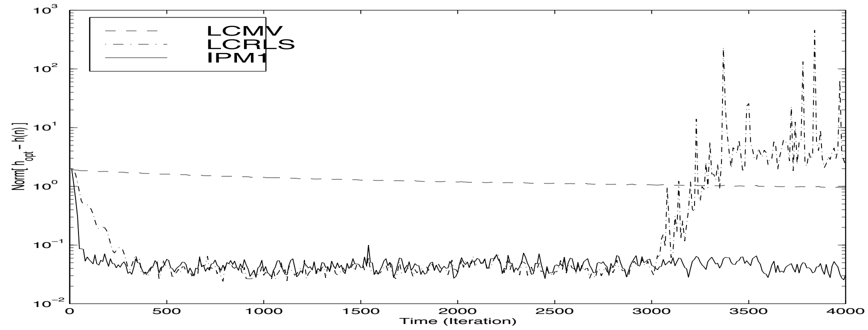


Figure 1: Mean least-squares error of h(n)

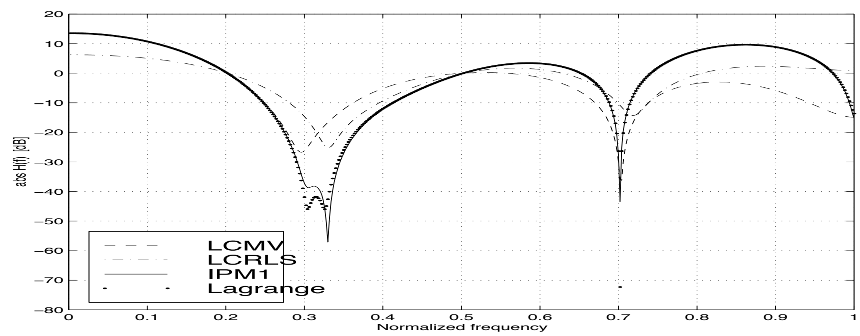


Figure 2: Frequency response of filters at Iteration 4000

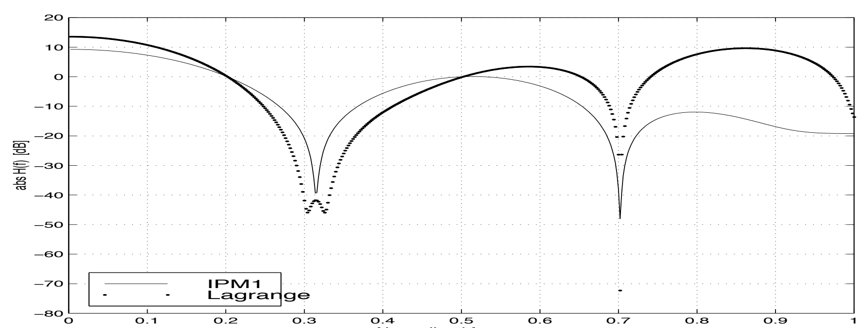


Figure 3: Mean-squared error of h(n) for IPM1 at Iteration 200,000 ($\lambda = 0.7$)

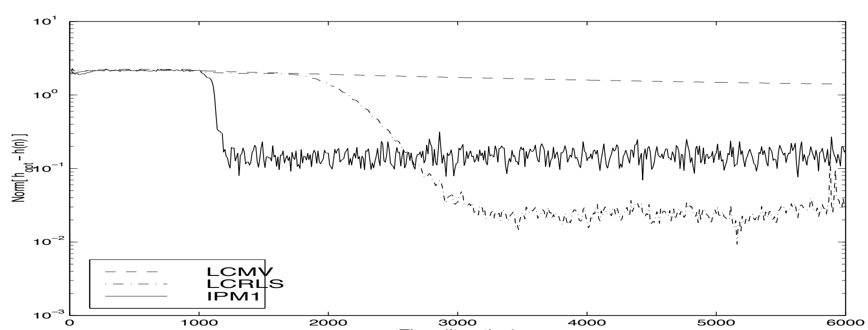


Figure 4: Abrupt change at iteration 1000