A Leakage Quasi-Newton Adaptation Algorithm

Fabiano T. Castoldi and Marcello L. R. de Campos

***Abstract*— This paper proposes an alternative view on adaptive filtering algorithm development and analysis. More than just rewriting objective functions minimized by the algorithms, the alternative approach explored in this article gives us extra tools for optimizing with respect to other parameters, for example the convergence factor μ.**

Adaptation algorithms are usually developed based either on a stochastic approximation of the gradient vector and Hessian matrix, or on a deterministic minimization of quadratic *a posteriori* output errors. Gradient-descent algorithms, such as the LMS (*Least Mean Squares*) algorithm and the QN (*Quasi Newton*) algorithm, are usually placed in the first group, whereas the RLS (*Recursive Least Squares*) algorithm is placed in the second group. Obviously these are just how algorithms are usually presented and analyzed; the RLS algorithm can also be seen as an stochastic approximation algorithm, and the LMS algorithm also does minimize a deterministic objective function. However, some descriptions of deterministic functions minimized by some algorithms, such as the LMS algorithm, offer very limited insight on its behavior. In this work we propose to shed new light onto known adaptation algorithms by means of describing their deterministic objective function as a quadratic norm of the coefficients, optionally subjected to equality constraints, which are functions of the output error. We show how this approach can be used to derive some LS-based and QN-based adaptation algorithms, such as the Leakage QN Algorithm mentioned in the title.

***Keywords*— Convex Optimization, Adaptation Algorithms, Adaptive signal processing**

I. INTRODUCTION

Many adaptation algorithms have been proposed in the past forty years, offering trade-offs in convergence speed, robustness, and computational complexity. Lack of robustness may be due to accumulation of quantization errors or due to loss of positive definiteness of the Hessian matrix caused by nonpersistently exciting input signals [1].

The RLS algorithm presents good convergence speed, but its robustness is not guaranteed unless we opt for a QR-decomposition implementation, or some other regularization scheme. Robust RLS-algorithm implementations with reduced computational complexity are usually based on QR decompositions, which are complex to implement and maintain [2]. There are other algorithms that have been developed based on known convex optimization methods, like the Quasi-Newton and IPLS (Interior Point Least Square) algorithms [1], [3]. These algorithms offer increased robustness at a cost of extra computational complexity, for they do not admit $O(N)$ implementations.

In this paper we present a different approach to the derivation of the conventional adaptation algorithms, whereby the objective function is described as a minimum-norm function subjected, or not, to constraints.

Fabiano T. Castoldi and Marcello L. R. de Campos, Programa de Engenharia Elétrica, COPPE/UFRJ, P.O.Box 68504, Rio de Janeiro, RJ, 21.945-970, Brazil. Emails: ftoldi@gmail.com and mcampos@ieee.org. This work was partially supported by CAPES and CNPq.

This paper is organized as follows. In Section II we present new derivation for some LS-based and the QN-based algorithms based on their least-squares deterministic objective function and the minimum-disturbance counterpart. In order to demonstrate the potential of those new algorithms, Section III provide some numerical simulations. Concluding remarks and some directions for further research are given in Section IV.

II. CONVEX OPTIMIZATION INTERPRETATION

A. Least-Squares Algorithm A

The RLS algorithm has certainly become one of the preferred alternatives to gradient-type algorithms in applications where fast convergence is needed. Although computationally complex and possibly unstable, the RLS algorithm provides excellent performance in terms of convergence speed and coefficient-error variance in steady state. The algorithm can be derived as a recursive implementation that minimizes the sum of the squares of the *a posteriori* output errors. One possible and general format of the recursive least squares objective function can be written as [2]

$$\xi_{A,n} = \sum_{i=1}^{n} \mu_i \left[d_i - \mathbf{x}_i^T \mathbf{w}_n\right]^2 \tag{1}$$

where d_i and $\mathbf{x}_i$ denote the desired response and input-signal vector pair at time instant i, and $\mathbf{w}_n$ denotes the coefficient vector at time instant n. The weights, μ_i, usually are positive scalars. The objective function $\xi_{A,n}$ is related to algorithm A at time instant n.

If one wants to obtain vector $\mathbf{w}_n$ that minimizes $\xi_{A,n}$ at time instant n, the following approach may be used:

$$\frac{\partial \xi_{A,n}}{\partial \mathbf{w}_n} = 0 \Longrightarrow \sum_{i=1}^{n} \mu_i d_i \mathbf{x}_i = \sum_{i=1}^{n} \mu_i \mathbf{x}_i \mathbf{x}_i^T \mathbf{w}_n$$

We may define the sample-based stochastic approximations of the input-signal autocorrelation matrix, $\mathbf{R}_{A,n}$, and input-reference-signal crosscorrelation vector, $\mathbf{p}_{A,n}$, as

$$\mathbf{R}_{A,n} = \sum_{i=1}^{n} \mu_i \mathbf{x}_i \mathbf{x}_i^T = \mathbf{R}_{A,n-1} + \mu_n \mathbf{x}_n \mathbf{x}_n^T \tag{2}$$

and

$$\mathbf{p}_{A,n} = \sum_{i=1}^{n} \mu_i d_i \mathbf{x}_i = \mathbf{p}_{A,n-1} + \mu_n d_n \mathbf{x}_n$$

respectively. The set of equations that solve the minimization problem can be rewritten as

$$\mathbf{R}_{A,n} \mathbf{w}_n = \mathbf{p}_{A,n}$$

which is in the form of Wiener-Hopf equations [4]. The coefficient vector that solves this set of linear equations is unique if and only if $\mathbf{R}_{A,n}$ has full rank, and it corresponds to a global minimum in the objective function surface if and only if $\mathbf{R}_{A,n}$ is positive definite. The latter should be the case for all persistently exciting signals of order M, where M is the dimension of the problem, i.e., $\mathbf{w}_n$ and $\mathbf{x}_n$ are $M \times 1$ vectors. The coefficient-vector solution to the minimization problem described by Eq. (1) is given by

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{e_n}{\frac{1}{\mu_n} + \tau_n}\mathbf{t}_n \quad (3)$$

where:

$$e_n = d_n - \mathbf{x}_n^T \mathbf{w}_{n-1} \quad (4)$$

$$\tau_n = \mathbf{x}_n^T \mathbf{R}_{A,n-1}^{-1} \mathbf{x}_n \quad (5)$$

$$\mathbf{t}_n = \mathbf{R}_{A,n-1}^{-1} \mathbf{x}_n \quad (6)$$

Here e_n is the *a priori* output error and the inversion of matrix $\mathbf{R}_{A,n}$ needed in Eqs. (5) and (6) can benefit from the recursive structure of Eq. (2) via Woodbury formula, also known as matrix-inversion lemma [4], which is given by

$$[A + BCD] = A^{-1} - A^{-1}B\left[C^{-1} + DAB\right]DA^{-1} \quad (7)$$

Using the matrix-inversion lemma given by Eq. (7) and making $A = \mathbf{R}_{n-1}$, $B = D^T = \mathbf{x}_n$ and $C = \mu_n$, the inversion of matrix in Eq. (2) is

$$\mathbf{R}_{A,n}^{-1} = \mathbf{R}_{A,n-1}^{-1} - \frac{\mathbf{t}_n \mathbf{t}_n^T}{\frac{1}{\mu_n} + \tau_n} \quad (8)$$

where $\mathbf{t}_n$ and τ_n are given by Eqs. (6) and (5), respectively.

One may quickly realize that Eq. (1), and consequently Eqs. (3)–(6), describe the conventional RLS algorithm when $\mu_i = 1$. Although general, formulation of the objective function as in Eq. (1) does not yield the exponentially weighted RLS algorithm, as described in [2] and [4].

1) A Minimum-Disturbance Description: An alternative objective function that yields the exact same algorithm as described by Eqs. (3)–(6) is presented below. The alternative objective function follows a minimum-disturbance objective given by some quadratic norm of the coefficient disturbance from iterations $n-1$ to n, but also takes into account the instantaneous squared *a posteriori* output error weighted by μ_n:

$$\xi_{A,n} = \|\mathbf{w}_n - \mathbf{w}_{n-1}\|^2_{\mathbf{R}_{A,n-1}} + \mu_n\left[d_n - \mathbf{w}_n^T\mathbf{x}_n\right]^2 \quad (9)$$

In the equation above, the quadratic norm is defined as

$$\|\mathbf{x}\|^2_{\mathbf{A}} = \mathbf{x}^T\mathbf{A}\mathbf{x}$$

for some vector $\mathbf{x}$, given any positive definite matrix $\mathbf{A}$.

This minimum-disturbance description has been used before to derive and analyze adaptation algorithms, e.g., [5]. Our approach and objectives in this paper, however, are slightly different.

By minimizing $\xi_{A,n}$ in Eq. (9) with respect to the filter coefficients $\mathbf{w}_n$, one can obtain the same recursive update as in Eq. (3):

$$\nabla\xi_{A,n} = 2\mathbf{R}_{A,n-1}\left[\mathbf{w}_n - \mathbf{w}_{n-1}\right] - 2\mu_n\left[d_n - \mathbf{w}_n^T\mathbf{x}_n\right]\mathbf{x}_n = 0$$

$$\left[\mathbf{R}_{A,n-1} + \mu_n\mathbf{x}_n^T\mathbf{x}_n\right]\mathbf{w}_n = \mathbf{R}_{A,n-1}\mathbf{w}_{n-1} + \mu_n d_n \mathbf{x}_n$$

$$\mathbf{R}_{A,n}\mathbf{w}_n = \left[\mathbf{R}_{A,n} - \mu_n\mathbf{x}_n^T\mathbf{x}_n\right]\mathbf{w}_{n-1} + \mu_n d_n\mathbf{x}_n$$

$$\mathbf{R}_{A,n}\mathbf{w}_n = \mathbf{R}_{A,n}\mathbf{w}_{n-1} + \mu_n e_n \mathbf{x}_n$$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu_n e_n \mathbf{R}_{A,n}^{-1}\mathbf{x}_n \quad (10)$$

In Eq. (10) one can use Eq. (8) to yield the same Eq. (3). The advantages of this alternative minimum-disturbance approach will be explored in some more detail in the following sections.

Eq. (1) presents an objective function $\xi_{A,n}$ which is a sum of squared *a posteriori* output errors calculated at point $\mathbf{w}_n$ and weighted by their corresponding μ_i. This strategy possess only limited ability to "forget" past information; at each time instant n, μ_n only acts upon the current *a posteriori* error $\varepsilon_n = d_n - \mathbf{x}_n^T\mathbf{w}_n$, as can be verified by Eq. (9). In the next section, a different function is proposed as objective for minimization.

B. Least-Squares Algorithm B

As an alternative objective function to that described in Eq. (1), a weighted least squares objective function can be defined, where at any time instant n the weight μ_n acts upon the whole sequence of past *a posteriori* errors, i.e.,

$$\xi_{B,n} = \sum_{i=1}^{n} \mu_{i,n}\left[d_i - \mathbf{x}_i^T\mathbf{w}_n\right]^2 \quad (11)$$

with

$$\mu_{i,n} = \prod_{j=i+1}^{n} \mu_j = \mu_n\ \mu_{i,n-1}$$

As usual, vector $\mathbf{w}_n$, which minimizes $\xi_{B,n}$, is obtained taking the first derivative of $\xi_{B,n}$ with respect to $\mathbf{w}_n$ and making it equal to zero, as follows:

$$\frac{\partial\xi_{B,n}}{\partial\mathbf{w}_n} = 0 \Longrightarrow \sum_{i=1}^{n}\mu_{i,n}d_i\mathbf{x}_i = \sum_{i=1}^{n}\mu_{i,n}\mathbf{x}_i\mathbf{x}_i^T\mathbf{w}_n$$

The stochastic approximations of the input-signal autocorrelation matrix and input-reference crosscorrelation vector are given by

$$\mathbf{R}_{B,n} = \sum_{i=1}^{n}\mu_{i,n}\mathbf{x}_i\mathbf{x}_i^T = \mu_n\mathbf{R}_{B,n-1} + \mathbf{x}_n\mathbf{x}_n^T \quad (12)$$

and

$$\mathbf{p}_{B,n} = \sum_{i=1}^{n}\mu_{i,n}d_i\mathbf{x}_i = \mu_n\mathbf{p}_{B,n-1} + d_n\mathbf{x}_n$$

respectively. One can readily spot the weighted RLS algorithm above, either from the recursive representations of $\mathbf{R}_{B,n}$ and $\mathbf{p}_{B,n}$, or from Eq. (11).

The coefficient-vector solution to the weighted least squares minimization problem described by Eq. (11) is given by

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{e_n}{\mu_n + \tau_n}\mathbf{t}_n \quad (13)$$

where e_n, τ_n, and $\mathbf{t}_n$ are defined as before in Eqs. (4)–(6), and $\mathbf{R}_{B,n}^{-1}$ is

$$\mathbf{R}_{B,n}^{-1} = \frac{1}{\mu_n}\left[\mathbf{R}_{B,n-1}^{-1} - \frac{\mathbf{t}_n\mathbf{t}_n^T}{\mu_n + \tau_n}\right] \quad (14)$$

The differences between Eqs. (1) and (11) are subtle, but nonetheless important to explain algorithm behavior and capabilities. For the latter, the "forgetting factor" at any time instant n has some control over the entire history of data pairs $\mathbf{x}_i$ and d_i. The exponentially weighted recursive least squares algorithm [4][2] fits nicely in this framework if we make $\mu_n = \lambda$, i.e., the RLS forgetting factor [4].

1) A Minimum-Disturbance Description: As in the previous case, the least squares algorithm with variable forgetting factor just described can also be alternatively formulated under the minimum-disturbance approach. In this case,

$$\xi_{B,n} = \mu_n \|\mathbf{w}_n - \mathbf{w}_{n-1}\|^2_{\mathbf{R}_{B,n-1}} + \left[d_n - \mathbf{x}_n^T \mathbf{w}_n\right]^2$$

The minimum-disturbance description of the objective function, in this case, shows that the "convergence factor" is applied to the quadratic norm of the coefficient disturbance from iterations $n-1$ to n, not to the squared *a posteriori* output error as before. This way of presenting the RLS algorithm is not the most common one, but has some resemblance with that presented in [5]. The derivation to obtain Eqs. (13) and (14) is similar to that carried out for Algorithm A.

C. Least-Squares Algorithm C

If we examine closely the minimum-disturbance descriptions of the algorithms presented in the previous sections, a natural, perhaps trivial, modification of the objective function is to make it a convex combination between minimum-coefficient disturbance and *a posteriori* output error, as follows:

$$\xi_{C,n} = (1-\mu_n)\|\mathbf{w}_n - \mathbf{w}_{n-1}\|^2_{\mathbf{R}_{C,n-1}} + \mu_n \left[d_n - \mathbf{x}_n^T \mathbf{w}_n\right]^2$$

If we make the first derivative of $\xi_{C,n}$ with respect to $\mathbf{w}_n$ equal to zero, we obtain

$$\frac{\partial \xi_{C,n}}{\partial \mathbf{w}_n} = 0$$

$$\Longrightarrow (1-\mu_n)\mathbf{R}_{C,n-1}(\mathbf{w}_n - \mathbf{w}_{n-1}) = \mu_n \mathbf{x}_n (d_n - \mathbf{x}_n^T \mathbf{w}_n)$$

If we define $\mathbf{R}_{C,n-1}\mathbf{w}_{n-1} = \mathbf{p}_{C,n-1}$ and

$$\mathbf{R}_{C,n} = (1-\mu_n)\mathbf{R}_{C,n-1} + \mu_n \mathbf{x}_n \mathbf{x}_n^T$$
$$\mathbf{p}_{C,n} = (1-\mu_n)\mathbf{p}_{C,n-1} + \mu_n d_n \mathbf{x}_n$$

then naturally $\mathbf{R}_{C,n}\mathbf{w}_n = \mathbf{p}_{C,n}$. Using the matrix-inversion lemma we have

$$\mathbf{R}^{-1}_{C,n} = \frac{1}{1-\mu_n}\left[\mathbf{R}^{-1}_{C,n-1} - \frac{\mathbf{t}_n \mathbf{t}_n^T}{\frac{1-\mu_n}{\mu_n} + \tau_n}\right] \quad (15)$$

and following the same deduction as in Algorithm A, one can obtain the update formula for the filter coeficients as

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{e_n}{\frac{1-\mu_n}{\mu_n} + \tau_n}\mathbf{t}_n \quad (16)$$

Interesting enough, this third option also becomes equivalent to the RLS algorithm when $1-\mu_n = \lambda < 1$, which has already been named LMS-Newton in [6]. Indeed, the same algorithm can be obtained if we minimize

$$\xi_{C,n} = \sum_{i=1}^{n} \prod_{j=i+1}^{n} (1-\mu_j)\mu_i \left[d_i - \mathbf{x}_i^T \mathbf{w}_n\right]^2 \quad (17)$$

with respect to $\mathbf{w}_n$.

D. On the Choice of μ

Although it can be stated generally that factor μ is a positive scalar that controls how past and present data is weighted, considerable research has been devoted in the past three decades to the study of "optimal" values for this parameter. Optimality criteria can be speed of convergence, misadjustment, robustness, to name a few. In the following sections we will touch the surface of the subject, pointing out how constraints can be included into the minimum-disturbance representation of objective functions. These constraints can normalize solutions such that they lie in the hyperplane of zero *a posteriori* output errors, or they can somehow improve robustness, for example.

E. Other Normalized Algorithms

From what has been discussed so far, it seems appropriate at this point to investigate other options of objective functions following the minimum disturbance approach. For the normalized least mean squares (NLMS) algorithm, the quadratic norm is taken with respect to the identity matrix and a constraint for zero *a posteriori* output error shall be enforced:

$$\xi_{NLMS,n} = \|\mathbf{w}_n - \mathbf{w}_{n-1}\|^2 \qquad \text{s.t.} \quad d_n = \mathbf{x}_n^T \mathbf{w}_n$$

The solution after minimizing $\xi_{NLMS,n}$ with respect to $\mathbf{w}_n$ is the NLMS algorithm. Using the technique of Lagrange multipliers, we have

$$\xi_{NLMS,n} = \|\mathbf{w}_n - \mathbf{w}_{n-1}\|^2 + \alpha\left[d_n - \mathbf{x}_n^T \mathbf{w}_n\right]$$

$$\nabla \xi_{NLMS,n} = 2\left[\mathbf{w}_n - \mathbf{w}_{n-1}\right] - \alpha \mathbf{x}_n = 0$$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{\alpha \mathbf{x}_n}{2} \quad (18)$$

Substituing this result in the restriction to find α, yields

$$d_n - \mathbf{x}_n^T\left[\mathbf{w}_{n-1} + \frac{\alpha \mathbf{x}_n}{2}\right] = 0$$

$$2e_n = \alpha \|\mathbf{x}_n\|^2$$

$$\alpha = \frac{2e_n}{\|\mathbf{x}_n\|^2}$$

Now, using this value of α in Eq. (18) to obtain the recursion formula for the filter coefficients, we have

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{e_n}{\|\mathbf{x}_n\|^2}\mathbf{x}_n$$

Any other positive matrix matrix would do in the quadratic norm above, not just the identity matrix. However, it seems natural that the identity matrix is as good as any other choice, if we have no knowledge about the statistics of the input signal.

Another normalized algorithm, in the sense that it also yields zero *a posteriori* error, is the QN algorithm proposed in [1]. This algorithm was developed based on the rank-one update of the Hessian matrix, and its good numerical properties are in part due to the fact that the condition

$$\mathbf{x}_n^T \mathbf{R}^{-1}_{QN,n} \mathbf{x}_n = \frac{1}{2}$$

is always satisfied. The minimum-disturbance description of the QN algorithm is given below:

$$\xi_{QN,n} = \|\mathbf{w}_n - \mathbf{w}_{n-1}\|^2_{\mathbf{R}_{QN,n-1}} \text{ s.t. } \begin{cases} d_n = \mathbf{x}_n^T \mathbf{w}_n \\ \mathbf{x}_n^T \mathbf{R}_{QN,n}^{-1} \mathbf{x}_n = 1/2 \end{cases} \quad (19)$$

After minimizing $\xi_{QN,n}$ with respect to $\mathbf{w}_n$, we obtain

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{e_n}{\tau_n}\mathbf{t}_n$$

and

$$\mathbf{R}_{QN,n}^{-1} = \mathbf{R}_{QN,n-1}^{-1} - \frac{1 - \frac{1}{2\tau_n}}{\tau_n}\mathbf{t}_n\mathbf{t}_n^T$$

or, equivalently,

$$\mathbf{R}_{QN,n} = \sum_{i=1}^{n} 2\left[1 - \frac{1}{2\tau_i}\right]\mathbf{x}_n\mathbf{x}_n^T$$

Normalization implies a 3dB penalty in the algorithm misadjustment [1]. It might be worth investigating what modification in $\xi_{QN,n}$ would be necessary in order to maintain a good numerical behavior, but at the same time allow a penalty to be applied to the normalization in order to let the solution wander away from the hyperplane given by $d_n - \mathbf{x}_n^T\mathbf{w}_n = 0$.

F. Leakage QN Algorithm

An objective function of a "leakage" quasi-Newton algorithm that preserves the good numerical properties of the original QN algorithm proposed in [1] can be constructed as

$$\xi_{LQN,n} = \mu_n\|\mathbf{w}_n - \mathbf{w}_{n-1}\|^2_{\mathbf{R}_{LQN,n-1}} + \left[d_n - \mathbf{x}_n^T\mathbf{w}_n\right]^2$$
$$\text{s.t.} \quad \mathbf{x}_n^T\mathbf{R}_{LQN,n}^{-1}\mathbf{x}_n = 1/2 \quad (20)$$

Minimization of the equation above with respect to the coefficient vector, $\mathbf{w}_n$, yields

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{e_n}{\mu_n + \tau_n}\mathbf{t}_n$$

with

$$\mathbf{R}_{LQN,n}^{-1} = \frac{1}{\mu_n}\left[\mathbf{R}_{LQN,n-1}^{-1} - \frac{\mathbf{t}_n\mathbf{t}_n^T}{\mu_n + \tau_n}\right] \quad (21)$$

as in Eq. (14). As the restriction imposed to this algorithm does not include the coefficient vector, $\mathbf{w}_n$, then the only difference between this algorithm and the algorithm presented in the subsection II-B is in the choice of the μ_n.

Applying the restriction $\mathbf{x}_n^T\mathbf{R}_{LQN,n}^{-1}\mathbf{x}_n = 1/2$ to the update formula of $\mathbf{R}_{LQN,n}^{-1}$, gives

$$\mathbf{x}_n^T\left\{\frac{1}{\mu_n}\left[\mathbf{R}_{LQN,n-1}^{-1} - \frac{\mathbf{t}_n\mathbf{t}_n^T}{\mu_n + \tau_n}\right]\right\}\mathbf{x}_n = 1/2$$

or, equivalently,

$$\frac{1}{\mu_n}\left[\frac{\tau_n(\mu_n + \tau_n) - \tau_n^2}{\mu_n + \tau_n}\right] = 1/2$$

Solving the equation above with respect to μ_n to find the algorithm step update, yields

$$\mu_n = \tau_n$$

III. SIMULATIONS

Simulations were carried out to test convergence of the algorithms only. As out main goal was to show a different perspective to adaptation-algorithm development and analysis, with possible generalization of known algorithms and improvements to their robustness characteristics, our simulations did not intend to compare algorithms in terms of their convergence speed or misadjustment, as is usually the case. In our simulations, an unknown system was to be identified by the adaptive filter and zero-mean white noise was added to the output signal of the system to form the reference signal. The signal-to-noise ratio (SNR) was constant and equal to 40dB for all simulations.

The unknown system was an FIR filter with transfer function given by

$$H(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} \quad (22)$$

normalized to unitary gain.

For each different algorithm, a sufficient order adaptive filter was used and the simulations were realized until convergence was reached. The input signal to the unknown system and to the adaptive filter was the same zero-mean white noise signal. The MSE (Mean Squared Error) shown in the figures were obtained after averaging over an ensemble of 100 simulations. In the simulations, the algorithms will be named after the section where they were presented.

Learning curves of algorithms A to C are shown in figure 1, where we adjust the value of μ_n to obtain the same misadjustment for all three algorithms.

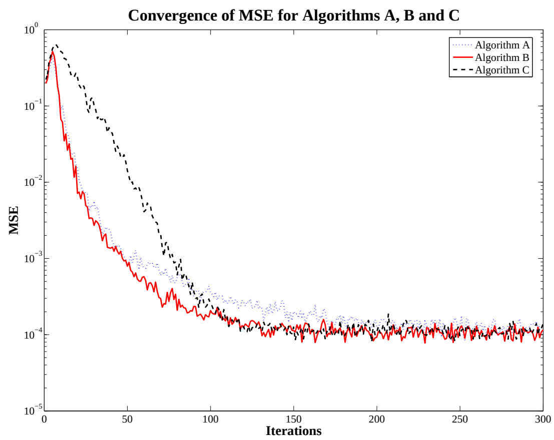

Fig. 1. Learning curves of Algorithms A, B and C.

Figure 2 shows an interesting comparison between the algorithms E (QN algorithm) and F (LQN algorithm), where it is possible to observe the effect of the normalization in the misadjustment. These two algorithms use variable values for μ_n, given by

$$\mu_n = \frac{1}{2\tau_n} \quad (23)$$

and

$$\mu_n = \tau_n \quad (24)$$

respectively.

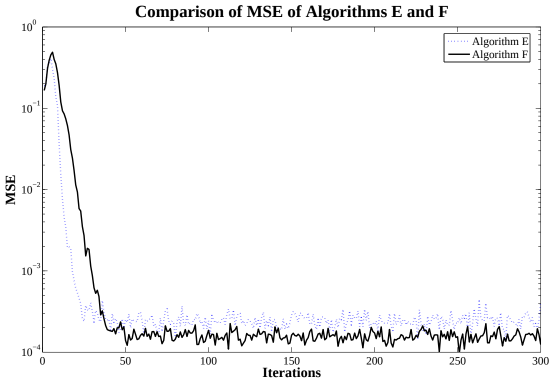

Fig. 2. Comparison of the misadjustment in MSE between QN algorithm and Leakage QN algorithm.

IV. CONCLUSIONS AND CRITIQUE

This work presented an alternative tool for the derivation of adaptation algorithms, via a coefficient-vector minimum-disturbance approach possibly combined with the squared *a posteriori* output error and equality constraints. Although the minimum-disturbance approach has been used before to present alternative derivations of the LMS algorithm and its variations, we have showed that a proper choice of the quadratic norm used to measure the *disturbance* of the coefficients can yield different LS-like or QN-like algorithms. Besides giving us tools to derive new algorithms, the approach described herein puts all algorithms under the same framework and provides us different insight on their expected behavior.

Perhaps the most interesting aspect of the framework just described is the presentation of an alternative deterministic objective function for LS-like algorithms that does not include n instances of the coefficient vector. Although a detail that might easily pass unnoticed, the alternative deterministic objective function allows us to optimize with respect to the convergence factor, μ_n, without the difficulty imposed by the implicit dependence of the n instances of $\mathbf{w}_n$ to μ_n. Although the derivation of adaptation algorithms and variable forgetting factors are subjects that may have already been explored to exhaustion, still our work offers a different perspective that may attract and amuse the expert and help the novice.

REFERÊNCIAS

[1] M. L. R. de Campos and A. Antoniou, "A new quasi-newton adaptive filtering algorithm," in *IEEE Transactions on Circuits and Systems*, November 1997, vol. 44, pp. 924–934.
[2] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Kluwer Academic Publishers, Norwell, Massachusetts, 2002.
[3] K. H. Afkhamie, Z.-Q. Luo, and K. M. Wong, "Adaptive linear filtering using interior point optimization techniques," in *IEEE Transactions on Signal Processing*, June 2000, vol. 48, pp. 1637–1647.
[4] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, 2nd edition, 1991.
[5] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, 2003.
[6] M. L. R. de Campos, P. S. R. Diniz, and A. Antoniou, "Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor," *IEEE Transactions of Signal Processing*, vol. 43, pp. 617–627, 1995.