Transient and steady-state MSE analysis of the IMPNLMS algorithm

Diego B. Haddad b,*, Mariane R. Petraglia a

a Federal University of Rio de Janeiro, EPPGI, COPPE, CP 68504, CEP 21945-970, Rio de Janeiro, Brazil
b CEFET, Nova Iguaçu, and Federal University of Rio de Janeiro, COPPE, Rio de Janeiro, Brazil

A R T I C L E   I N F O
Article history:
Available online 9 July 2014

Keywords:
Adaptive filters
Convergence analysis
Sparse impulse response identification
Proportionate normalized least mean squares algorithm

A B S T R A C T
Several techniques have been proposed in the literature to accelerate the convergence of adaptive algorithms for the identification of sparse impulse responses (i.e., with energy concentrated in a few coefficients). Among these techniques, the improved μ-law proportionate normalized least mean squares (IMPNLMS) algorithm is one of the most effective. This paper presents an accurate transient analysis of this algorithm and derives an estimate of its steady-state MSE, without requiring the assumption of white Gaussian input signals.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

This paper focuses on the class of supervised adaptive algorithms that seek the identification of sparse impulse responses, common in various acoustic, chemical and seismic processes, as well as in wireless communications channels [1,2]. Here, a sequence is considered sparse if most of its elements are close to zero, which implies a concept of sparsity weaker than that usually employed in numerical analysis [3].

There is no mathematical difficulty in applying supervised algorithms to identify such responses, but some practical problems may arise, the main one being a slow convergence. In this paper we investigate the convergence behavior of algorithms of the PNLS family [4] to accelerate the convergence of the identification method, analyzing the performance evolution of one of its most successful algorithms (the so-called IMPNLMS [5]) and obtaining an estimate of its mean-square error in steady state.

If the adaptive filter has length L, we can define the input vector x k in terms of the input signal x(k) as

\[ x_k = [x(k) \hspace{1em} x(k-1) \hspace{1em} \cdots \hspace{1em} x(k-L+1)]^T. \] (1)

Although the input signal can be colored, the vast majority of the transient analysis of PNLS-type algorithms found in the literature assume that it is white [4,6–8]. Violation of this assumption results in substantial differences in the convergence of the algorithms.

In the k-th iteration, the adaptive weight vector \( \hat{h}_k \) is expressed as

\[ \hat{h}_k = [\hat{h}_k(0) \hspace{1em} \hat{h}_k(1) \hspace{1em} \cdots \hspace{1em} \hat{h}_k(L-1)]. \] (2)

and the optimal response to be provided by the adaptive algorithm is \( d(k) = h x_k \). In this work, we assume that the length of the filter impulse response h to be identified is equal to or less than L and that the uncertainty on the desired response measurement can be modeled by an additive white Gaussian noise ν(k), which is independent of the input signal and has variance \( \sigma^2 \).

The unpredictability of this noise prevents its removal, and hence the adaptive system has only access to \( d(k) = d(k) + \nu(k) \). The adaptive algorithm should change the parameters to minimize a cost function dependent on the measured error defined as

\[ e(k) = d(k) - y(k) = [h - \hat{h}_k] x_k + \nu(k), \] (3)

where \( y(k) = \hat{h}_k x(k) \) is the adaptive filter output signal at instant k. Fig. 1 illustrates the structure of a typical supervised adaptive identification algorithm.

The process of minimizing the cost function determines the characteristics of the adaptive learning system. Among the various cost functions found in the literature, the most popular is the squared error \( e^2(k) \) [9–11], which can be interpreted as an instantaneous estimate of the mean square error (MSE). The corresponding adaptation algorithm, known as Least-Mean-Square (LMS) algorithm, employs the steepest descent optimization method. Its normalized version (the NLMS algorithm), accelerates the convergence rate by varying the learning factor along the iterations,

\[ \mu = \frac{1}{\text{constant}}. \]

1 The weight vectors are defined as row vectors, while all other vectors are column vectors.

* Corresponding author.
E-mail addresses: diego@pads.ufrj.br (D.B. Haddad), mariane@pads.ufrj.br (M.R. Petraglia).
avoiding, through the normalization, that eventually vectors $x_k$ with high modulus amplify the noise in the adaptive process. The update equation of the NLMS algorithm is

$$
\tilde{h}_{k+1} = \tilde{h}_k + \beta \frac{x_k e(k)}{\delta + x_k^T x_k},
$$

where $\delta$ is a constant slightly larger than zero to avoid divisions by zero and $\beta$ is the step-size or learning factor.

Recently several adaptive algorithms have been proposed specifically for sparse response identification systems. Such algorithms aim at overcoming the slow convergence of the NLMS in sparse configurations [1,12] through an uneven distribution of updating energy over the coefficients of $\tilde{h}_k$, with larger learning factors assigned to the coefficients of greater magnitude. This strategy can be interpreted as a cooperation established by a central resource administrator, which gives more prominent updates for the coefficients of greater magnitude.

In this context, the PNLMS algorithm increases the factor $\beta$ corresponding to $\tilde{h}_k(n)$ proportionally to its magnitude. The elements of $\tilde{h}_k$ that are farther away from zero will have larger updates than those of smaller magnitude. The algorithm also performs a regularization for small amplitude signals [4].

### 2. Adaptive algorithms for the identification of sparse responses

In this section we briefly present some of the main contributions in adaptive identification of sparse impulse responses.

It is very common to use the NLMS for the adaptation of high-order adaptive filters, such as in echo cancellation [4]. One of the first proposed alternatives consisted of using filters with fewer adaptive coefficients than the length of the impulse response, by only updating the subsets of coefficients that corresponded to the dispersive regions [13,14]. One of the great advantages of this strategy lies in the substantial reduction in computational cost.

Another possibility of accelerating the adaptation convergence in the context of sparse impulse responses, when the NLMS has a sub-optimal performance, is the distribution of the learning factor $\beta$ through the coefficients, as explained above. The first such proposal was the PNLMS (Proportional Normalized Least-Mean-Squares) algorithm, derived for echo cancellation [4]. All the algorithms studied in this paper are derived from the PNLMS, and for that reason we say that they belong to the family of the PNLMS algorithms.

For sparse systems, the PNLMS algorithm presents faster initial convergence than does the NLMS. However, the convergence rate is dramatically reduced after the initial period and is slower than that of the NLMS for non-sparse impulse responses [5]. For this reason, the PNLMS+ algorithm [15] adopts switching between PNLMS and NLMS algorithms in order to reduce this degradation in non-sparse configurations. In [16], an approximation of the optimal step-size control factors is proposed in order to circumvent this drawback of the PNLMS algorithm. Instead of adjusting the adaptation step-size proportionally to the magnitude of the estimated filter coefficient, the resulting algorithm employs the logarithm of these magnitudes. In order to reduce the computational cost, the logarithmic function is approximated by a piecewise linear function, leading to the $\mu$-law proportionate NLMS (MPNLMS) algorithm.

The above approaches have the disadvantage of requiring sparseness of the impulse response to be identified for fast convergence, which is not always the case. In [5] such problem is mitigated by employing a measure of the sparseness of the system impulse response, resulting in the improved MPNLMS (IMPNLMS) algorithm shown in Table 1. The function $\hat{\delta}_k$ estimates the degree of sparseness based on the available estimated impulse response at each iteration. Such function assumes values in the interval $[0, 1]$, approaching 1 when the impulse response is sparse and 0 when it is dispersive. The conversion of $\hat{\delta}_k$ to the domain of the parameter $\alpha(k)$ was arbitrated by simulations [5]. The piecewise linear function

$$
F(\hat{\delta}_k(n)) = \begin{cases} 
400|\hat{\delta}_k(n)|, & 0 < \hat{\delta}_k(n) < 0.005 \\
8.51|\hat{\delta}_k(n)| + 1.96, & \text{otherwise}
\end{cases},
$$

which approximates the logarithmic function [16], is adopted in the update of the step-size control factors $\hat{g}_k(i)$.

Among others, alternative strategies (not explored in this paper) for the identification of sparse responses consist of using an approximation of $l_0$-norm or $l_1$-norm of the weight vector to obtain a more accurate sparseness measure [17–20] and the use of Krylov subspace [6].

### 3. Transient analysis of the IMPNLMS algorithm

The theoretical estimation of the mean square error convergence of an adaptive algorithm eliminates the need of Monte Carlo averaging, among other advantages already well acknowledged in the literature. In this section, we derive recursive equations that describe in a reasonably accurate form the evolution of the MSE along the iterations.

In all analyses of PNLMS-type algorithms found in the literature, it is assumed that the input signal is white [7,8,21]. The violation of this hypothesis makes the algorithm convergence much slower, which disagrees with the analytical results. Therefore, in the benefit of generality, our analysis imposes no constraint on the input signal. We focus on the analysis of the IMPNLMS.

The equations of interest here are (see Table 1):

$$
\hat{\delta}_k = \frac{L}{L - \sqrt{L}} \left( 1 - \frac{\sum_{j=0}^{L-1} \hat{h}_k(j)}{\sqrt{L \sum_{j=0}^{L-1} \hat{h}_k(j)^2}} \right),
$$

### Table 1

<table>
<thead>
<tr>
<th>IMPNLMS algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialization (typical values)</strong></td>
</tr>
<tr>
<td>$\delta = 0.01$, $\varepsilon = 0.001$, $\beta = 0.25$, $\lambda = 0.1$</td>
</tr>
<tr>
<td>$\hat{\delta}(-1) = 0.96$</td>
</tr>
<tr>
<td>$\hat{\delta}_0 = [\hat{\delta}_0(0) \hat{\delta}_0(1) \cdots \hat{\delta}_0(L-1)] = [0]$</td>
</tr>
<tr>
<td><strong>Processing and adaptation</strong></td>
</tr>
<tr>
<td>For $k = 0, 1, 2, \cdots$, $x_k = [x(k) x(k-1) \cdots x(k-L+1)]^T$</td>
</tr>
<tr>
<td>$y(k) = \tilde{h}_k x_k$</td>
</tr>
<tr>
<td>$e(k) = d(k) - y(k)$</td>
</tr>
<tr>
<td>$\hat{\delta}<em>k = \frac{1}{L} \left( \frac{\sum</em>{j=0}^{L-1}</td>
</tr>
<tr>
<td>$x(k) = (-1 - \lambda</td>
</tr>
<tr>
<td>$x(k) = 2x(k-1)$</td>
</tr>
<tr>
<td>For $k = 0, 1, 2, \cdots, L - 1$, $\hat{g}<em>k(l) = \frac{L - \alpha(l)}{L} + \frac{\lambda \alpha(l)}{L \sum</em>{j=0}^{L-1}</td>
</tr>
<tr>
<td>End For</td>
</tr>
<tr>
<td>$F_k = \text{diag}(</td>
</tr>
<tr>
<td>$\tilde{h}_{k+1} = \tilde{h}_k + \beta x_k e(k)$</td>
</tr>
<tr>
<td>$\hat{\delta}<em>{k+1} = \frac{1}{L} \sum</em>{j=0}^{L-1} \hat{g}_k(j)$</td>
</tr>
<tr>
<td>End For</td>
</tr>
</tbody>
</table>
\[ \xi(k) = (1 - \lambda)\xi(k - 1) + \lambda \xi_{h(i)}, \] (7)

\[ \alpha(k) = 2\alpha(k - 1), \] (8)

\[ g_k(i) = \frac{1 - \alpha(k)}{2L} + \frac{(1 + \alpha(k))F(\hat{\xi}_h(i))}{2\sum_{j=0}^{L-1} F(\hat{\xi}_h(j)) + \epsilon}, \] (9)

\[ \hat{\xi}_{h+1} = \hat{\xi}_h + \beta \sum_{k=1}^{N} \Gamma_k \xi_k + \delta. \] (10)

To simplify the expected value calculations, we define the deviation of the \( j \)-th element of \( \hat{\xi}_h \) as

\[ \hat{\xi}_k(j) = \hat{\xi}_h(j) - h(j), \] (11)

and \( \hat{\xi}_k = [\hat{\xi}_k(0) \hat{\xi}_k(1) \cdots \hat{\xi}_k(L - 1)]. \)

The computation of the following terms is required for the MSE estimates at each iteration:\footnote{In related literature, the assumption of white input signals makes it necessary only to estimate the second order terms \( \mathbb{E}[g_k^2(i)] \), as our analysis does not use this restriction, we must also calculate the cross-terms.}

- i) \( \mathbb{E}[\xi_h(i)] \); ii) \( \mathbb{E}[\xi(k)] \); iii) \( \mathbb{E}[\alpha(k)] \); iv) \( \mathbb{E}[g_k(i)] \); v) \( \mathbb{E}[\hat{\xi}_{k+1} \ | \ \hat{\xi}_k(i)] \); vi) \( \mathbb{E}[\sum_{j=0}^{L-1} g_k(j)\xi_k(j)] \).

To obtain i)-vi), it is necessary to calculate the intermediate terms: vii) \( \mathbb{E}[\hat{\xi}_h(j)] \); viii) \( \mathbb{E}[F(\hat{\xi}_h(j))] \); ix) \( \mathbb{E}[\sum_{j=0}^{L-1} g_k(j)\xi_k(j)] \).

For some of the above expected values, analytical forms cannot be derived, which forces us to use simplifying assumptions in order to find recursive equations for the MSE estimate at each iteration. The effects of employing such assumptions on the analysis terms that are affected by them are experimentally evaluated in Section 5 and shown in Figs. 2-6.

The main assumptions used in our transient MSE analysis are:

- Hypothesis I: the input signal \( x(k) \) is Gaussian, stationary, has zero mean, and is uncorrelated with the measurement noise \( v(k) \), which also is a zero mean process. This assumption is justified by the use of the linear model, albeit unknown, and by the fact that the order of the optimal Wiener filter coincides with that of the unknown system [7].

- Hypothesis II-a: The expected value of the ratio between random variables that depend on the coefficients of the adaptive filter is approximated by the ratio between the expected values of these variables, that is, \( \mathbb{E}[X/Y] \approx \mathbb{E}[X]/\mathbb{E}[Y] \).

- Hypothesis II-b: The expected value of the ratio between random variables that depend on the coefficients of the adaptive filter is approximated by the expected value of the numerator multiplied by the expected value of the inverse of the denominator,\footnote{An important denominator in our analysis is \( (\sum_{j=0}^{L-1} g_k(j)^2(k - j) + \delta)^n \), which, being constant (or satisfying \( L \gg 2\sum_{j=0}^{L-1} \mathbb{E}[g_k(j)^2] \) [22]), validates the approach of the Hypothesis II-a. The calculation of this term by Hypothesis II-b is more accurate.} that is, \( \mathbb{E}[X/Y] \approx \mathbb{E}[X]/\mathbb{E}[Y] \).

- Hypothesis III:

\[ \mathbb{E} \left[ \left( \sum_{j=0}^{L-1} \hat{\xi}_h(j)^2 \right) \right] \approx \sqrt{\mathbb{E} \left[ \left( \sum_{j=0}^{L-1} \hat{\xi}_h(j)^2 \right) \right]}. \]

- Hypothesis IV: \( \mathbb{E}[\alpha(k)F(\hat{\xi}_h(i))] \approx \mathbb{E}[\alpha(k)] \mathbb{E}[F(\hat{\xi}_h(i))] \), where \( F \) is the piecewise linear function of Eq. (5).

- Hypothesis V: The coefficients of the LMS estimator act as a low-pass filter, and hence we assume that \( \hat{\xi}_h(i) \) and \( g_k(i) \) vary slowly with respect to \( x(k - j) \). Implicitly, this means that the learning factor is sufficiently small, so that we can make the following approximations:

\[ \mathbb{E}[g_k(i)x^n(k - j)] \approx \mathbb{E}[g_k(i)] \mathbb{E}[x^n(k - j)], \] (12)

\[ \mathbb{E}[g_k(i)^n \hat{\xi}_h(i)^m] \approx \mathbb{E}[g_k(i)]^n \mathbb{E}[\hat{\xi}_h(i)^m], \] (13)

and

\[ \mathbb{E}[\hat{\xi}_h(i)^n x^n(k - j)] \approx \mathbb{E}[\hat{\xi}_h(i)] \mathbb{E}[x^n(k - j)]. \] (14)

- Hypothesis VI: the coefficients \( \hat{\xi}_h(i) \) have a Gaussian distribution,\footnote{Due to the recursive algorithm, we can evolve the central limit theorem to justify this hypothesis, at least after a number of iterations.} with mean \( \mu_k(i) \) and variance \( \sigma_k^2(i) \).

- Hypothesis VII: \( \mathbb{E}[g_k(i)\hat{\xi}_h(i)] \approx 0 \mathbb{E}[g_k(i)], \) for \( i \neq l \).

Next, we underline the main steps for solving the expected value of each term:

- i) Term \( \mathbb{E}[\xi_h(i)] \).

From Eq. (6), we have

\[ \mathbb{E}[\xi_h(i)] = \frac{L}{L - \sqrt{\epsilon}} \left( 1 - \mathbb{E} \left[ \frac{\sum_{j=0}^{L-1} \hat{\xi}_h(j)}{\sqrt{\sum_{j=0}^{L-1} \hat{\xi}_h^2(j)}} \right] \right). \] (15)

Using Hypotheses II-a and III, we can write:

\[ \mathbb{E}[\xi_h(i)] \approx \frac{L}{L - \sqrt{\epsilon}} \left( 1 - \mathbb{E} \left[ \frac{\sum_{j=0}^{L-1} \hat{\xi}_h(j)}{\sqrt{\sum_{j=0}^{L-1} \hat{\xi}_h^2(j)}} \right] \right). \] (16)

where \( \mathbb{E}[\hat{\xi}_h(j)] \) is Term vii), which will be obtained by using Hypothesis VI, and \( \mathbb{E}[\hat{\xi}_h^2(j)] \) can be easily derived from Term vi).

- ii) Term \( \mathbb{E}[\alpha(k)] \).

From Eq. (7), we obtain

\[ \mathbb{E}[\alpha(k)] = (1 - \lambda) \mathbb{E}[\xi(k - 1)] + \lambda \mathbb{E}[\xi_h(i)]. \]

- iii) Term \( \mathbb{E}[g_k(i)] \).

From Eq. (8), we find

\[ \mathbb{E}[g_k(i)] = 2\mathbb{E}[\xi(k)] - 1. \]

- iv) Term \( \mathbb{E}[\hat{\xi}_h(i)] \).

From Eq. (9) and Hypotheses II-a and IV, it follows that

\[ \mathbb{E}[\hat{\xi}_h(i)] \approx \frac{1 - \mathbb{E}[\alpha(k)]}{2L} + \frac{(1 + \mathbb{E}[\alpha(k)])\mathbb{E}[F(\hat{\xi}_h(i))]}{2\sum_{j=0}^{L-1} \mathbb{E}[F(\hat{\xi}_h(j))] + \epsilon}. \] (17)

where the term \( \mathbb{E}[F(\hat{\xi}_h(j))] \) is derived subsequently, using Hypothesis VI.

- v) Term \( \mathbb{E}[\sum_{j=0}^{L-1} \hat{\xi}_h(j)] \).

Let \( \mathbb{R}_k = \mathbb{E}[\xi_k(i) \xi_k(i)] \). Using Hypotheses II-b and V we obtain

\[ \mathbb{E}[\sum_{j=0}^{L-1} \hat{\xi}_h(j)] \approx \mathbb{E}[\xi_h(i)] - \mathbb{E}[\xi_k(i)] \mathbb{R}_k \mathbb{E}[\xi_h(i)]. \] (18)

where \( \mathbb{E}[\xi_k(i)] = \mathbb{E}[1/\sum_{j=0}^{L-1} g_k(j)^2(k - j) + \delta] \). A more accurate derivation of \( \mathbb{E}[\xi_k(i)] \) by Hypothesis II-b will be carried out in the determination of Term vi).

- vi) Term \( \mathbb{E}[\sum_{j=0}^{L-1} \hat{\xi}_h(j)] \).

The update of the \( i \)-th coefficient of \( \hat{\xi}_h \) is given by:

\[ \hat{\xi}_h(i) = \hat{\xi}_h(i) + \frac{\beta g_k(i)x(k - j)v(k)}{\sum_{j=0}^{L-1} g_k(j)^2} \hat{\xi}_h(j) + \frac{\delta}{\sum_{j=0}^{L-1} g_k(j)^2} \hat{\xi}_h(j) - h(j) \] (19)
From the above recursion and Hypotheses II-b, V and VII, we obtain:
\[
E[\mathcal{Z}_{k+1}(i)\mathcal{Z}_{k+1}(l)] = E[\mathcal{Z}_k(i)\mathcal{Z}_k(l)] - \beta E_g.1 g_k(0) \sum_{j=0}^{L-1} E[\mathcal{Z}_k(j)\mathcal{Z}_k(i)] r(i-j) + \beta^2 E_g.2 E[\mathcal{g}_k(i)] E[\mathcal{g}_k(l)] r(i-l) \sigma_k^2 - \beta E_g.1 E[\mathcal{g}_k(i)] \sum_{n=0}^{L-1} E[\mathcal{Z}_n(n)\mathcal{Z}_k(i)] r(i-n) + \beta^2 E_g.2 E[\mathcal{g}_k(i)] \sum_{n=0}^{L-1} \sum_{j=0}^{L-1} E[\mathcal{Z}_n(j)\mathcal{Z}_k(n)] r(i-n) \times r_{i,j,n,}\]
where \(r(n) = E[x(n)x(k-n)]\) and \(r_{i,j,n,} = E[x(n-i)x(k-j)x(k-l)x(-n)]\). Using Hypothesis I, \(r_{i,j,n,}\) are fourth order moments of Gaussian variables, which can be easily obtained.

vii) Term \(E[\hat{\mathcal{h}}_k(i)]\).
From Hypothesis VI, \(\hat{\mathcal{h}}_k(i)\) is Gaussian, with mean \(\mu_k(i)\) and variance \(\sigma_k^2\), which can be obtained from the evaluation of Terms v) and vi). Thus, the probability density function (pdf) of \(\hat{\mathcal{h}}_k(i)\) is given by:
\[
f(\hat{\mathcal{h}}_k(i)) = \frac{1}{\sqrt{2\pi\sigma_k(i)}} e^{-\frac{(\hat{\mathcal{h}}_k(i) - \mu_k(i))^2}{2\sigma_k^2(i)}} U(\hat{\mathcal{h}}_k(i)),
\]
where \(U(\cdot)\) is the step function.

The following definite integrals will be useful in the derivations that follow:
\[
\Phi_0(a, b, \mu_k(i), \sigma_k(i)) = \int_a^b e^{-\frac{(\hat{\mathcal{h}}_k(i) - \mu_k(i))^2}{2\sigma_k^2(i)}} d\hat{\mathcal{h}}_k(i) = \sigma_k(i) \sqrt{\frac{\pi}{2}} \left[ \text{erfc} \left( \frac{a - \mu_k(i)}{\sqrt{2}\sigma_k(i)} \right) - \text{erfc} \left( \frac{b - \mu_k(i)}{\sqrt{2}\sigma_k(i)} \right) \right],
\]
\[
\Phi_1(a, b, \mu_k(i), \sigma_k(i)) = \int_a^b e^{-\frac{(\hat{\mathcal{h}}_k(i) - \mu_k(i))^2}{2\sigma_k^2(i)}} d\hat{\mathcal{h}}_k(i) = \sigma_k(i)^2 \left[ e^{-\frac{(a-\mu_k(i))^2}{2\sigma_k^2(i)}} - e^{-\frac{(b-\mu_k(i))^2}{2\sigma_k^2(i)}} \right] + \mu_k(i) \sigma_k(i) \sqrt{\frac{\pi}{2}} \left[ \text{erfc} \left( \frac{a - \mu_k(i)}{\sqrt{2}\sigma_k(i)} \right) - \text{erfc} \left( \frac{b - \mu_k(i)}{\sqrt{2}\sigma_k(i)} \right) \right],
\]
where \(\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-v^2} dv\).

With the pdf of \(\hat{\mathcal{h}}_k(i)\) and the integral \(\Phi_1\) defined above,\(^5\) the expression \(E[\hat{\mathcal{h}}_k(i)]\) can be finally determined as:
\[
E[\hat{\mathcal{h}}_k(i)] = \int_0^\infty \hat{\mathcal{h}}_k(i) f(\hat{\mathcal{h}}_k(i)) d\hat{\mathcal{h}}_k(i) = \frac{\Phi_1(0, \infty, \mu_k(i), \sigma_k(i))}{\sqrt{2\pi\sigma_k(i)}} + \frac{\Phi_1(0, \infty, -\mu_k(i), \sigma_k(i))}{\sqrt{2\pi\sigma_k(i)}}.
\]

viii) Term \(E[F(\hat{\mathcal{h}}_k(i))]\).
Using statistical average and Eqs. (5) and (21), we obtain:
\[
E[F(\hat{\mathcal{h}}_k(i))] = \frac{1}{2\pi\sigma_k(i)} \left[ 400\Phi_1(5 \times 10^{-3}, \mu, \sigma) + 400\Phi_1(5 \times 10^{-3}, -\mu, \sigma) + 8.51\Phi_1(5 \times 10^{-3}, \infty, \mu, \sigma) + 8.51\Phi_1(5 \times 10^{-3}, -\infty, -\mu, \sigma) + 1.96\Phi_0(5 \times 10^{-3}, \infty, \mu, \sigma) + 1.96\Phi_0(5 \times 10^{-3}, -\infty, -\mu, \sigma) \right].
\]

ix) Term \(E_{g,n} = E[1/(\sum_{j=0}^{L-1} \mathcal{g}_k(i) x^2(k-j) + \delta)^n]\). \(E_{g,n}\) can be estimated by using Hypothesis II-a as follows:
\[
E_{g,n} = \frac{1}{\sum_{j=0}^{L-1} \mathcal{g}_k(i) x^2(k-j) + \delta} \approx \left( \sum_{j=0}^{L-1} \mathcal{g}_k(i) \right)^{-n} \approx \frac{1}{(\sigma_k^2 + \delta)^n}.
\]

where \(\sigma_k^2\) is the variance of \(x(k)\). The above step was identified as the most critical in the accuracy of the analysis, especially when the adaptive filters have short lengths, since in these cases the variance of the denominator is large. According to the definition of \(\mathcal{h}_k\), even a short impulse response can have a maximum degree of sparseness (when only one of its elements is nonzero), which could motivate the use of IMPNLM in this configuration.

A more accurate estimate of \(E_{g,n}\) can be developed by employing the techniques of \(22,26-28\) and \(29\), for \(n = 1, 2\), as detailed below.

As the regularization parameter \(\delta\) assumes values close to zero, the expansion of \(E_{g,n}\) by a Taylor series can be approximated as:
\[
E_{g,1} = E \left[ \sum_{j=0}^{L-1} \mathcal{g}_k(i) x^2(k-j) + \delta \right] \approx \sum_{j=0}^{L-1} \mathcal{g}_k(i) x^2(k-j) - \delta \left( \sum_{j=0}^{L-1} \mathcal{g}_k(i) x^2(k-j) \right)^2.
\]
\[
E_{g,2} = E \left[ \frac{1}{(\sum_{j=0}^{L-1} \mathcal{g}_k(i) x^2(k-j) + \delta)^2} \right] \approx \frac{1}{(\sum_{j=0}^{L-1} \mathcal{g}_k(i) x^2(k-j))^2} - 2\delta \frac{1}{(\sum_{j=0}^{L-1} \mathcal{g}_k(i) x^2(k-j))^3}.
\]

Defining \(\tilde{E}_{g,n} = E[1/(\mathcal{X}^2 \Gamma_0)^n]\), we observe that \(E_{g,n}\) and \(\tilde{E}_{g,n}\) are related as:
\(^6\) Reference \(30\), in a similar context, proposes this approach for the calculation of \(E_{g,1}\).
\[ E_{g,1} \approx E_{g,1} - \delta E_{g,2}, \]  
\[ E_{g,2} \approx E_{g,2} - 2\delta E_{g,3}. \]  
\[ \tilde{E}_{g,n} \approx E \left[ \frac{1}{(\mathbf{x}'_n E[\Gamma_k|x_n])^n} \right]. \]  

The components of \( E[\Gamma_k] \) have been calculated recursively in Term iv. 

The pdf \( f(x_k) \) is known from Hypothesis I, resulting in:

\[ E \left[ \frac{1}{(\mathbf{x}'_n E[\Gamma_k|x_n])^n} \right] = \frac{1}{(2\pi)^{L/2} \sqrt{\det(R_k)}} \int \cdots \int (\mathbf{x}'_n E[\Gamma_k|x_n])^n e^{-\mathbf{x}'_n E[\Gamma_k|x_n] \mathbf{x} - \frac{1}{2} \mathbf{x}'_n \mathbf{R}_k \mathbf{x}} d\mathbf{x}. \]  

Letting \( \Psi_0(\omega) \):

\[ \Psi_0(\omega) = \frac{1}{(2\pi)^{L/2} \sqrt{\det(R_k)}} \int \cdots \int (\mathbf{x}'_n E[\Gamma_k|x_n])^n e^{-\mathbf{x}'_n E[\Gamma_k|x_n] \mathbf{x} - \frac{1}{2} \mathbf{x}'_n \mathbf{R}_k \mathbf{x}} d\mathbf{x}. \]  

we can write

\[ E \left[ \frac{1}{(\mathbf{x}'_n E[\Gamma_k|x_n])^n} \right] = \Psi_0(0) \]

\[ = \frac{1}{n^L} \int \cdots \int \frac{\partial^n \Psi_0(\omega)}{\partial \omega^n} d\omega \cdots d\omega |_{\omega=0}. \]  

Deriving \( n \) times the function \( \Psi_0(\omega) \) with respect to \( \omega \) and employing properties of the Gaussian pdf, we obtain

\[ \tilde{E}_{g,n} = \frac{1}{n^L} \int \cdots \int \frac{(-1)^n}{\sqrt{\det(I + 2\omega \mathbf{R}_k \Gamma_k)} d\omega \cdots d\omega |_{\omega=0}.} \]  

The above expression requires the calculation of a hyper-elliptic integral, which, except in cases where \( L \) is small, does not have analytical solution. Therefore, we have to resort to approximations. Letting \( \lambda_i \) be the eigenvalues of \( E[\Gamma_k] R_k \), we can rewrite Eq. (35) as

\[ \tilde{E}_{g,n} = \frac{1}{n^L} \int \cdots \int \frac{(-1)^n}{(\prod_{i=1}^L (1 + 2\omega \lambda_i))^{1/2}} d\omega \cdots d\omega |_{\omega=0}.} \]  

where \( \omega_\ell = -1/2 \lambda_\ell \). As in the approach reported in [24], we replace adjacent roots \( \omega_{2q-1} \) and \( \omega_{2q} \) of the polynomial by a single root \( \omega_q^{(1)} \), with multiplicity 2, obtained by the geometric mean of the adjacent roots, that is,

\[ \omega_q^{(1)} = -\sqrt{\omega_{2q-1} \omega_{2q}}. \]  

With this approximation, we obtain:

\[ \tilde{E}_{g,n} \approx \frac{1}{n^L} \int \cdots \int \frac{1}{(\omega - \omega_1) \cdots (\omega - \omega_L)} d\omega \cdots d\omega |_{\omega=0}.} \]  

The partial fraction expansion of \( \varphi(\omega) \) yields

\[ \tilde{E}_{g,n} \approx \frac{1}{n^L} \int \cdots \int \frac{A_1}{(\omega - \omega_1) \cdots (\omega - \omega_L)} d\omega \cdots d\omega |_{\omega=0}.} \]

where

\[ A_q = \frac{1}{\prod_{i=1}^L (\omega_\ell - \omega_{\ell+1})}. \]  

For \( n = 1, 2 \) and 3, we find

\[ \tilde{E}_{g,1} \approx \frac{-1}{2L \lambda_1} \sum_{i=1}^{L/2} A_1 \ln(-\omega_i), \]  

\[ \tilde{E}_{g,2} \approx \frac{1}{2L \lambda_1} \sum_{i=1}^{L/2} A_1 [\omega_i - \omega_i \ln(-\omega_i)], \]  

\[ \tilde{E}_{g,3} \approx \frac{1}{2L \lambda_1} \sum_{i=1}^{L/2} A_1 [\omega_i^2 \ln(-\omega_i)^2]. \]  

4. Steady-state MSE estimation of PNLMS-type algorithms

The MSE value obtained after convergence is an important performance measure of adaptive algorithms. Using the technique of energy conservation [31], we obtain a theoretical formula for the steady-state MSE of the PNLMS family.

The following hypotheses were assumed to allow the derivation of a closed form expression for the steady-state MSE:

- **Hypothesis I**: The adaptive algorithm does not diverge, and remains stable when \( k \to \infty \).
- **Hypothesis II**: \( v(k) \) has zero mean and is independent of \( x_j, \forall j \).
Hypothesis III: $1/|x_k|^2$ and the a priori error $e_a(k)$ (defined in Eq. (45)) are uncorrelated random variables when $k \to \infty$.

Hypothesis IV: The order of the adaptive filter is equal to or larger than the order of the filter to be identified, and during transient the learning factor associated with each coefficient is not zero.

The update equation (Eq. (10)) of the algorithms of the PNLMS family, expressed in terms of the deviations $\hat{z}_k$ and $\hat{z}_{k+1}$ and neglecting the regularization parameter $\delta$, yields

$$ \hat{z}_{k+1} = \hat{z}_k + \frac{\beta x_k^T \Gamma_k e(k)}{x_k^T \Gamma_k x_k}. $$

(44)

Defining the a priori and a posteriori errors

$$ e_a(k) = -\hat{z}_k x_k $$

and

$$ e_p(k) = -\hat{z}_k x_{k+1}, $$

(45)

(46)

respectively, and after multiplying the terms in Eq. (44) on the right by $x_k$, we find

$$ e_p(k) = e_a(k) - \beta e(k), $$

(47)

from which it follows that

$$ e(k) = \frac{e_a(k) - e_p(k)}{\beta}. $$

(48)

Substituting Eq. (48) into Eq. (44), we obtain

$$ \hat{z}_{k+1} + x_k^T \Gamma_k e_p(k) = \hat{z}_k + x_k^T \Gamma_k e_a(k). $$

(49)

The energy of the left hand side of Eq. (49) is given by

$$ E_e = \left\| \hat{z}_{k+1} + x_k^T \Gamma_k e_p(k) \right\|^2 - \frac{x_k^T \Gamma_k x_k}{\beta} e_p(k)^2 + \frac{1}{\beta} e_a(k)^2. $$

Letting $k \to \infty$ and taking the expected value, we obtain

$$ \text{E}[E_e] = \text{E}\left[\|\hat{z}_k\|^2\right] - \frac{1}{\beta} \text{E}[e_a(k)^2] + \frac{1}{\beta} \text{E}[e_p(k)^2]. $$

(50)

A similar procedure applied to the right hand side of Eq. (49) gives

$$ \text{E}[E_d] = \text{E}\left[\|\hat{z}_k\|^2\right] - \frac{1}{\beta} \text{E}[e_a(k)^2] + \frac{1}{\beta} \text{E}[e_p(k)^2]. $$

(51)

Equating the energies of the left and right hand sides of Eq. (49) and using Hypothesis I, we can write

$$ \lim_{k \to \infty} \frac{\text{E}[e_a(k)]}{\|x_k\|^2} = \lim_{k \to \infty} \frac{\text{E}[e_p(k)]}{\|x_k\|^2}. $$

(52)

Writing $e(k) = e_a(k) + v(k)$, from Eq. (48) results

$$ e_a(k) = (1 - \beta^2) e_a^2(k) - 2\beta(1 - \beta) e_a(k) v(k) + \beta^2 v^2(k), $$

(53)

which, used in Eq. (52), yields

$$ \lim_{k \to \infty} \frac{\text{E}[e_a(k)]}{\|x_k\|^2} = \lim_{k \to \infty} \left\{ (1 - \beta^2) \text{E}\left[\frac{e_a^2(k)}{|x_k|^2}\right] - 2\beta(1 - \beta) \text{E}\left[\frac{e_a(k) v(k)}{|x_k|^2}\right] + \beta^2 \text{E}\left[\frac{v^2(k)}{|x_k|^2}\right] \right\}. $$

(54)

Using Hypothesis II, we obtain

$$ \lim_{k \to \infty} \frac{\text{E}[e_a(k)]}{\|x_k\|^2} = \lim_{k \to \infty} \left\{ (1 - \beta^2) \text{E}\left[\frac{e_a^2(k)}{|x_k|^2}\right] + \beta^2 \sigma_v^2 \text{E}\left[\frac{1}{|x_k|^2}\right] \right\}, $$

(55)

where $\sigma_v^2$ is the variance of $v(k)$. Under Hypothesis III, we observe that $\text{E}[e_a^2(k)/\|x_k\|^2] = \text{E}[e_a^2(k)]\text{E}[1/\|x_k\|^2]$, and consequently

$$ \lim_{k \to \infty} \frac{\text{E}[e_a(k)]}{\|x_k\|^2} = \frac{\beta \sigma_v^2}{2 - \beta}. $$

(56)

Using Hypothesis IV, we observe that $\lim_{k \to \infty} \text{E}[e_a^2(k)]$ is the excess mean-square error (the MSE above the noise variance) of the algorithm in steady-state [31], and therefore

$$ \lim_{k \to \infty} \frac{\text{E}[e^2(k)]}{\|x_k\|^2} = \sigma_v^2 + \frac{\beta \sigma_v^2}{2 - \beta} = \frac{\sigma_v^2}{1 - \beta}. $$

(57)

The result in Eq. (56) coincides with that presented in [4]. However, our derivation has two main advantages: it is simpler and uses fewer hypotheses. In [4], two additional hypotheses were used: $x_k$ is Gaussian and $\beta \ll 1$ or $g_k(j) \ll 1, \forall j$. Our derivation shows that these assumptions are not necessary. The restrictions imposed by Hypotheses I and IV on the learning factors and on $g_k(j)$ are less restrictive than those required in [4].

Eq. (57) indicates that the steady-state MSE of any algorithm of the PNLMS family is virtually independent of the energy distribution performed by the matrix $\Gamma$. Different choices of this matrix influence significantly only the convergence rate of the MSE.

The main recursive equations for the transient and steady-state analysis of the IMPNLMS algorithms are listed in Table 2. For initialization, we chose $E[h_0(i)] = \mu_0(i) = 0$ and $E[h_0(i)] = \delta_1$, where $\delta_1$ is a very small value.

5. Simulation results

5.1. Transient analysis

To illustrate the increased accuracy of the MSE transient analysis of the IMPNLMS algorithm, the identification of five different transfer functions was carried out. The transfer functions of the ITU-T Recommendation G.168 for echo cancellation [32] were used. Tables 3 and 4 show the settings for each simulated case. Table 3 presents the algorithm parameters and channel impulse response configurations, whereas Table 4 contains the input signal $x(k)$ and measurement noise $v(k)$ specifications. In Cases 1 to 7, $x(k)$ was obtained by applying a white Gaussian noise with unit variance to the filters of Table 4, while in Case 8, $x(k)$ was obtained by filtering a uniform white noise of zero mean and unit variance. The measurement noise was assumed white Gaussian with variance given in Table 4.

Figs. 2 through 6 compare the averages of the original terms and their respective approximations obtained by resorting to Hypotheses III-VII assumed in the theoretical analysis, for the first

9. The assumption $\beta \ll 1$ is used in the transient analysis only.
Table 2
Main recursive equations for the transient and steady-state analysis of the IMPNLMS algorithm.

<table>
<thead>
<tr>
<th>Transient analysis</th>
<th>Steady-state analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ E[x_k] = \frac{1}{\lambda} \left[ 1 - \sum_{j=0}^{\infty} E[x_j</td>
<td>k] \right] ]</td>
</tr>
<tr>
<td>[ E[x(k)] = (1 - \lambda)E[x(k-1)] + 2E[x_k] ]</td>
<td></td>
</tr>
<tr>
<td>[ E[x_k] = \frac{1}{\lambda} \left[ 1 - \sum_{j=0}^{\infty} E[x_j</td>
<td>k] \right] ]</td>
</tr>
<tr>
<td>[ E[x_k] = \frac{1}{\lambda} \left[ 1 - \sum_{j=0}^{\infty} E[x_j</td>
<td>k] \right] ]</td>
</tr>
<tr>
<td>[ E[x_{k+1}] = E[x_k] + \beta E[x_k] \Gamma_k ]</td>
<td></td>
</tr>
<tr>
<td>[ E[x_{k+1}] = E[x_k] + \beta E[x_k] \Gamma_k ]</td>
<td></td>
</tr>
<tr>
<td>[ E[x_{k+1}] = E[x_k] - \beta E[x_k] \Gamma_k ]</td>
<td></td>
</tr>
<tr>
<td>[ \Phi(a, b, \mu_k, \sigma_k(i)) = \sigma^2_k(i) \left( \frac{\mu_k}{\sqrt{2\pi} \sigma_k(i)} \right) ]</td>
<td></td>
</tr>
<tr>
<td>[ E[\hat{h}(0)] = \Phi(0, \infty, \mu, \sigma) ]</td>
<td></td>
</tr>
<tr>
<td>[ E[\hat{h}(i)] = \Phi(0, \infty, \mu, \sigma) ]</td>
<td></td>
</tr>
<tr>
<td>[ E[\hat{h}(i)] = \Phi(0, \infty, \mu, \sigma) ]</td>
<td></td>
</tr>
<tr>
<td>[ E[\hat{h}(i)] = \Phi(0, \infty, \mu, \sigma) ]</td>
<td></td>
</tr>
<tr>
<td>[ E[\hat{h}(i)] = \Phi(0, \infty, \mu, \sigma) ]</td>
<td></td>
</tr>
</tbody>
</table>

Table 3
Configurations tested in simulations. In all cases, \( \delta = 10^{-2}, \epsilon = 10^{-3}, \lambda = 10^{-4} \) and \( \xi(-1) = 0.96 \). Only \( L \) samples, from \( k_1 \) to \( k_f \), of each impulse response were used.

<table>
<thead>
<tr>
<th>Case</th>
<th>Model number</th>
<th>( L )</th>
<th>( \beta )</th>
<th>( k_1 )</th>
<th>( k_f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>16</td>
<td>0.25</td>
<td>20</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>20</td>
<td>0.25</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>10</td>
<td>0.25</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>25</td>
<td>0.2</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>30</td>
<td>0.2</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>21</td>
<td>0.95</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>12</td>
<td>0.25</td>
<td>35</td>
<td>46</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>31</td>
<td>0.2</td>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 4
Input signal coloring filter and measurement noise variance.

<table>
<thead>
<tr>
<th>Case</th>
<th>( \sigma^2 )</th>
<th>Coloring filter (equivalent MATLAB command)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 10^{-6} )</td>
<td>fir2([0 0.25 0.5 0.75 1]; [1 0.5 0 0])</td>
</tr>
<tr>
<td>2</td>
<td>( 10^{-5} )</td>
<td>fir2([0 0.25 0.5 0.75 1]; [1 0.5 0 0])</td>
</tr>
<tr>
<td>3</td>
<td>( 10^{-4} )</td>
<td>fir2([0 0.25 0.5 0.75 1]; [1 0.5 0 0])</td>
</tr>
<tr>
<td>4</td>
<td>( 10^{-3} )</td>
<td>fir2([0 0.25 0.5 0.75 1]; [1 0.5 0 0])</td>
</tr>
<tr>
<td>5</td>
<td>( 10^{-2} )</td>
<td>fir2([0 0.25 0.5 0.75 1]; [1 0.5 0 0])</td>
</tr>
<tr>
<td>6</td>
<td>( 10^{-1} )</td>
<td>fir2([0 0.25 0.5 0.75 1]; [1 0.5 0 0])</td>
</tr>
<tr>
<td>7</td>
<td>( 10^{0} )</td>
<td>fir2([0 0.25 0.5 0.75 1]; [1 0.5 0 0])</td>
</tr>
<tr>
<td>8</td>
<td>( 10^{1} )</td>
<td>fir2([0 0.25 0.5 0.75 1]; [1 0.5 0 0])</td>
</tr>
</tbody>
</table>

configuration of Tables 3 and 4. All results were achieved from 10,000 Monte Carlo averages. These figures show a reasonable good agreement between theory and the corresponding Monte Carlo averages, validating the hypotheses.

In the proposed analysis, \( E_{2.2} \) (Eq. (30)) can be estimated with or without the approximation \( E_{2.2} \approx E_{1.1}^2 \). To verify the effect of this approximation, two approaches were tested: without the ap-
proximation (Analysis Method 1) and with the proposed approach (Analysis Method 2). Figs. 7–14 compare the theoretical and experimental evolutions of the mean square error. Note that method 1 performs slightly better than method 2. As method 1 is computationally more expensive than method 2 (which requires the
Fig. 5. Probability density function, estimated by Gaussian kernels, of adaptive coefficients $\hat{h}_k(i)$ (in blue). (a) $k = 10$ and $i = 29$; (b) $k = 200$ and $i = 29$; (c) $k = 1000$ and $i = 29$; (d) $k = 10$ and $i = 31$; (e) $k = 200$ and $i = 31$; (f) $k = 1000$ and $i = 31$. The ideal coefficient values are in dashed red lines. In green, we present a Gaussian distribution with mean and standard deviation identical to the samples used to estimate the probability density function. Hypothesis VI assumes a Gaussian distribution for these coefficients. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. Experimental comparison of terms $E[|g_k(i)|^2]$ (in blue) and its approximations, derived from Hypothesis VII, $E[|\hat{g}_k(i)|^2]$ (in red). (a) $i = 5$; (b) $i = 10$; (c) $i = 15$; (d) $i = 20$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 7. MSE evolution (in dB) for Case 1.

Fig. 8. MSE evolution (in dB) for Case 2.

Fig. 9. MSE evolution (in dB) for Case 3.

Fig. 10. MSE evolution (in dB) for Case 4.

Fig. 11. MSE evolution (in dB) for Case 5.
calculation of the additional term $\tilde{E}_{2,3}$ of Eq. (43), the best computational cost × accuracy trade-off is obtained with method 2. Since the length of the adaptive filter is small, the standard method [21] for the transient analysis (which applies the hypothesis $E[A/B] \approx E[A]/E[B]$) is rather inaccurate. A much slower MSE convergence is obtained for colored input signals.

The configuration of Case 6 violates Hypothesis V of small $\beta$, which is used in the transient analysis. A larger difference between the theoretical and simulated results is observed in this case. However, method 2 still gives a reasonable estimate. Case 7 shows the use of a much lower SNR than those of the other cases. Even in this setting, method 1 can properly describe the evolution of the MSE, and method 2 provides a better steady-state estimate than does the standard analysis method. Case 8 uses a filtered uniform white noise (instead of a Gaussian noise), violating Hypothesis I. This case illustrates the fact that the proposed analysis technique is capable of providing a reasonable estimate of the transient behavior even when the input signals are not Gaussian.

5.2. Steady-state MSE analysis

Using model 2 of [32] and varying $\beta$ of the IMPNLMS algorithm, we calculate the MSE from Eq. (57) and from time averaging over 10,000 iterations after convergence. In this test, we set the input signal as unit-variance white Gaussian, $\sigma_n^2 = 10^{-6}$, $\delta = 10^{-4}$, $\epsilon = 10^{-3}$, $\lambda = 10^{-3}$ and $\xi(-1) = 0.96$. Fig. 15 shows close agreement between experimental and theoretical results.

6. Conclusions

In this work, we developed a theoretical analysis of the transient evolution of the MSE of the IMPNLMS algorithm. The proposed approach is not limited to white Gaussian input signals, and the approximations used were verified experimentally. A theoretical analysis of the steady-state MSE of the algorithms of the PNLS family was also presented. Such analysis employed the energy conservation technique, assuming less restrictive hypotheses than the previous ones reported elsewhere, resulting in a closed form expression for the steady-state MSE. Experimental results revealed that the proposed analysis techniques have a higher degree of accuracy than the analysis previously reported in the literature so far, to the best of our knowledge.

Acknowledgments

This work was supported in part by CNPq, Brazil.

References


Mariane R. Petraglia received the B.Sc. degree in electronic engineering from the Federal University of Rio de Janeiro, Brazil, in 1985, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of California, Santa Barbara, in 1988 and 1991, respectively. From 1992 to 1993, she was with the Department of Electrical Engineering at the Catholic University of Rio de Janeiro, Brazil. Since 1993, she has been with the Department of Electronic Engineering and with the Program of Electrical Engineering, COPPE, at the Federal University of Rio de Janeiro, where she is presently an Associate Professor. From March 2001 to February 2002, she was a Visiting Researcher with the Adaptive Systems Laboratory, at the University of California, Los Angeles. Since January 2014 she has been a Visiting Researcher at University of Southern California. Dr. Petraglia is a member of Tau Beta Pi. She served as an Associate Editor for the IEEE Transactions on Signal Processing from 2004 to 2007. Her research interests are in adaptive signal processing, blind source separation, multirate systems, and image processing.

Diego B. Haddad received the B.Sc. degree in telecommunications engineering from the State University of Rio de Janeiro, Brazil, in 2005, and the M.Sc. and D.Sc. degrees in electronic engineering at COPPE (Federal University of Rio de Janeiro), Brazil, in 2008 and 2013, respectively. Since 2008, he has been with Federal Center for Technological Education (CEFET), Brazil. His current research interests include adaptive signal processing, blind source separation, and pattern recognition.