ALTERNATIVE APPROACH FOR COMPUTING THE ACTIVATION FACTOR OF THE PNLMS ALGORITHM

Francisco das C. de Souza, Orlando J. Tobias, Rui Seara

LINSE – Circuits and Signal Processing Laboratory
Electrical Engineering Dept., UFSC, 88040-900, Brazil
{fsouza, orlando, seara}@linse.ufsc.br

Dennis R. Morgan

Bell Laboratories, Alcatel-Lucent
Murray Hill, NJ 07974-0636, USA
drrm@bell-labs.com

ABSTRACT

This paper presents a proportionate normalized least-mean-square (PNLMS) algorithm using an individual activation factor for each adaptive filter coefficient. Such strategy is used instead of a global activation factor as in the standard PNLMS algorithm. The proposed individual activation factors, determined in terms of the corresponding adaptive filter coefficients, lead to a better distribution of the adaptation energy over the filter coefficients than the standard PNLMS does. Thereby, for impulse responses exhibiting high sparseness, the proposed algorithm achieves faster convergence, outperforming both the PNLMS and improved PNLMS (IPNLMS) algorithms.

1. INTRODUCTION

Sparse impulse responses are encountered in many real-world applications, such as communications, acoustics, and seismic and chemical processes [1], [2], [3]. For this class of impulse responses, classical adaptive algorithms using the same step-size value for all filter coefficients, such as the normalized least-mean-square (NLMS) algorithm, converge slowly. To overcome this drawback, some algorithms exploiting the sparse nature of the impulse response have been proposed. The proportionate NLMS (PNLMS) algorithm [4] is one of these algorithms in which each filter coefficient is updated proportionally to its magnitude, resulting in higher convergence speed. However, the PNLMS algorithm presents some performance degradation as the sparseness decreases [5], [6]. Improved versions of the PNLMS algorithm aiming to deal with impulse responses exhibiting medium sparseness are the PNLMS++ [5] and improved PNLMS (IPNLMS) [6]. Nevertheless, these algorithms do not provide the same fast initial convergence obtained with the PNLMS for impulse responses having high sparseness [7]. A version of the PNLMS algorithm that takes into account the sparseness variation of the plant is the sparseness controlled PNLMS (SC-PNLMS) [8]. This algorithm performs well for both very high sparseness and medium dispersion; however, such performance is obtained at the expense of higher computational complexity with respect to the PNLMS.

The standard PNLMS algorithm performance depends on some predefined parameters controlling proportionality and initialization [1]. Thus, a central point is how to set suitable values for these parameters, since they impact the algorithm convergence speed. Such parameters are related to an algorithm variable, termed the activation factor, having the task to prevent the adaptive coefficients from stalling when their magnitudes are zero or are significantly smaller than the largest one. In the standard PNLMS algorithm, the activation factor is common to all coefficients, computed sample-by-sample, and depends on the instantaneous infinity-norm of the adaptive filter coefficient vector. This way to compute the activation factor leads to a gain distribution between the adaptive filter coefficients not entirely in line with the concept of proportionality, which is the desired attribute of the PNLMS algorithm. Hence, this work revisits the PNLMS algorithm for devising a new way to determine the activation factor, aiming to improve the algorithm performance. In contrast to the standard PNLMS algorithm, the new way to compute the activation factor has the following characteristics:

i) An individual activation factor is used for each adaptive filter coefficient.
ii) Each individual activation factor is computed in terms of the corresponding coefficient magnitude.
iii) The individual activation factors do not rely on the proportionality and initialization parameters, since they are no longer in the proposed formulation.

As a consequence, the convergence features of the proposed algorithm are significantly improved. Since there are now individual activation factors for each coefficient, the new algorithm version is named individual activation factor PNLMS (IAF-PNLMS).

For impulse responses having high sparseness, numerical simulations show that the proposed approach has faster convergence as well as faster response to perturbations of the system plant than both the PNLMS and IPNLMS algorithms.

2. STANDARD PNLMS ALGORITHM

2.1 Algorithm Formulation

The standard PNLMS algorithm is formulated by the following set of equations [1], [6]:

Coefficient update $(N \times 1)$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu \mathbf{G}(n)\, e(n) \mathbf{x}(n)}{\mathbf{x}^{\mathrm{T}}(n) \mathbf{G}(n) \mathbf{x}(n) + \varepsilon} \tag{1}$$

Error signal

$$e(n) = d(n) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n) + z(n) \tag{2}$$

Gain distribution matrix $(N \times N)$

$$\mathbf{G}(n) = \mathrm{diag}[g_1(\mathrm{n})\ g_2(\mathrm{n}) \cdots g_N(\mathrm{n})] \tag{3}$$

Individual gain

$$g_i(n) = \frac{\phi_i(n)}{\sum_{i=1}^{N} \phi_i(n)} \tag{4}$$

Proportionality function

$$\phi_i(n) = \max\left[f(n),\ |w_i(n)| \right] \tag{5}$$

Activation factor

$$f(n) = \rho \max\left[\delta,\ \|\mathbf{w}(n)\|_\infty \right] \tag{6}$$

This work was supported in part by the National Council for Scientific and Technological Development (CNPq) and the Committee for Postgraduate Courses in Higher Education (CAPES).

where $0<\mu<2$ is the step-size parameter, $\varepsilon>0$ is a regularization parameter (preventing division by zero and stabilizing the solution), and $\|\cdot\|_\infty$ is the infinity-norm. Variable $d(n)$ denotes the desired signal and $z(n)$ is a zero-mean i.i.d. measurement noise with variance σ_z^2 and uncorrelated with any other signal in the system. The input vector is $\mathbf{x}(n)=[x(n)\ x(n-1)\cdots x(n-N+1)]^{\mathrm{T}}$ and the adaptive filter vector of dimension N is $\mathbf{w}(n)=[w_1(n)\ w_2(n)\cdots w_N(n)]^{\mathrm{T}}$. Matrix $\mathbf{G}(n)$ distributes the gains between the coefficients, depending on their magnitude, with $g_i(n)$ governing the individual step-size adjustment. Activation factor $f(n)$ given in (6) depends on the adaptive filter coefficient vector as well as on ρ and δ, which are the proportionality (or activation) and initialization parameters, respectively. The initialization parameter permits starting the adaptation process at $n=0$, when all filter coefficients are initialized to zero. The proportionality (or activation) parameter prevents an individual coefficient from freezing when its magnitude is much smaller than the largest coefficient magnitude [5], [6].

2.2 Algorithm Discussion and Performance

In this section, the algorithm behaviour is discussed, aiming to get some insight to motivate an improved version of the PNLMS algorithm. To this end, the central point is to study the effect of the activation factor $f(n)$ and its associated parameters (proportionality and initialization) on the algorithm behaviour.

From (4) and (5), $g_i(n)$ is rewritten as

$$g_i(n)=\frac{1}{c(n)}\max\left[f(n),\ |w_i(n)|\right] \tag{7}$$

with

$$c(n)=\sum_{i=1}^{N}\phi_i(n) \tag{8}$$

being a common variable for all $g_i(n)$. Now, by analyzing (7), some important definitions are stated:

i) **Gain for inactive coefficients** $g^{\text{inactive}}(n)$. If $f(n)>|w_i(n)|$, the ith coefficient $w_i(n)$ is inactive and its associated gain, given by (7), is expressed as

$$g^{\text{inactive}}(n)=\frac{1}{c(n)}f(n). \tag{9}$$

Since activation factor $f(n)$ is common to all filter coefficients, a minimum and common gain $g^{\text{inactive}}(n)$ is assigned to all inactive coefficients. This is an undesirable feature, as will be shown further.

ii) **Gain for active coefficients** $g_i^{\text{active}}(n)$. If $f(n)\le|w_i(n)|$, the ith coefficient $w_i(n)$ is active and its associated gain, obtained from (7), is

$$g_i^{\text{active}}(n)=\frac{1}{c(n)}|w_i(n)|. \tag{10}$$

Here, each gain $g_i^{\text{active}}(n)$ is associated with the magnitude of the active coefficient $w_i(n)$, being the essence of the PNLMS algorithm.

Now, based on the above definitions (i) and (ii), let us analyze the total gain distributed over the filter coefficients at each iteration, which can be represented by the trace of matrix $\mathbf{G}(n)$. Thus,

$$\mathrm{tr}\left[\mathbf{G}(n)\right]=\frac{N-N_{\text{active}}}{c(n)}f(n)+\sum_{i\in A}g_i(n) \tag{11}$$

where N_{active} is the number of active coefficients and A is the set of indices associated with their positions. In (11), the first r.h.s. term is the total gain distributed over the inactive coefficients and the second, corresponds to the total gain distributed over the active ones. Note that a decrease in $f(n)$, for instance, implies an increase in the second r.h.s. term of (11), since $\mathrm{tr}\left[\mathbf{G}(n)\right]$ is always constant and equal to 1 from (3) and (4), meaning that the gains associated with the active coefficients are also dependent on $f(n)$. Therefore, we conclude that the activation factor $f(n)$ affects the gains assigned to both active and inactive coefficients. Hence, $f(n)$ plays an important role in the algorithm gain distribution. Here, we point out that in the formulation of the standard PNLMS algorithm (used by Benesty and Gay in [6]), the trace of matrix $\mathbf{G}(n)$ is equal to 1, whereas in [1] and [7], the trace of $\mathbf{G}(n)$ is set equal to N. However, both formulations are equivalent and we choose the former.

In the following, we study in detail the algorithm performance with respect the activation factor. To this end, Monte Carlo (MC) simulations (average of 100 independent runs) of the PNLMS algorithm are carried out for a system identification problem. The scenario for all numerical simulations consists of a sparse impulse response with $N=100$ coefficients, presented in [2], having its active coefficient values equal to $\{0.1,\ 1.0,\ -0.5,\ 0.1\}$ located at positions $\{1, 30, 35, 85\}$, respectively. To quantify the sparseness of this impulse response, we use a common measure, based on the relationship between the 1- and 2-norm of the impulse response, defined as [3], [9]

$$S(\mathbf{p})\triangleq\frac{N}{N-\sqrt{N}}\left(1-\frac{\|\mathbf{p}\|_1}{\sqrt{N}\|\mathbf{p}\|_2}\right) \tag{12}$$

where the N-dimensional vector $\mathbf{p}=[p_1\ p_2\cdots p_N]^{\mathrm{T}}$ represents the plant impulse response, and $\|\mathbf{p}\|_1$ and $\|\mathbf{p}\|_2$ are, respectively, 1- and 2-norm of $\mathbf{p}$. The measure $S(\mathbf{p})$ ranges from 0 (sparseness degree of a uniform filter) to 1 (sparseness degree of a Dirac filter). By using (12), the sparseness degree of the above impulse response is calculated as $S(\mathbf{p})=0.9435$. The input signal is a correlated unity-variance AR(2) process given by

$$x(n)=0.4x(n-1)-0.4x(n-2)+v(n) \tag{13}$$

where $v(n)$ is white noise with variance $\sigma_v^2=0.77$ and the eigenvalue spread of the autocorrelation matrix of the input vector is $\chi=10$. The measurement noise $z(n)$ is white with variance $\sigma_z^2=10^{-3}$ (SNR $=30$dB). To evaluate the algorithm performance, we use the normalized misalignment measure (in dB), given by [3]

$$\kappa(n)=10\log_{10}\frac{\|\mathbf{p}-\mathbf{w}(n)\|_2^2}{\|\mathbf{p}\|_2^2}. \tag{14}$$

Aiming to show the PNLMS algorithm dependence on the activation factor, some misalignment curves are shown in Figure 1 for $f(n)$ given by (6), $\mu=0.5$, $\delta=0.01$, and using ρ equal to 0.01, 0.05, and 0.50. In addition, a fourth numerical simulation, now using a constant activation factor $f(n)=0.001$, manually adjusted for maximizing the convergence speed, is also included in Figure 1. Note from the figure how the algorithm convergence is affected by both the activation factor and parameter ρ. By using $f(n)$ given in (6) and reducing the value of ρ, the convergence speed increases. For the case $f(n) = 0.001$, faster algorithm convergence is obtained, since the algorithm behaves in a more proportional manner, i.e., less gain per iteration is assigned to inactive coefficients. Therefore, from the curves of Figure 1, an interesting insight can be drawn: the activation factor given by (6) is not the best way to obtain a suitable gain distribution.

To assess the algorithm gain distribution, we define the following figures of merit:

Total gain distribution over L iterations

$$\theta_i = \sum_{n=0}^{L-1} g_i(n) \qquad (15)$$

Average of θ_i over the inactive coefficients

$$\theta_{\text{mean}}^{\text{inactive}} = \frac{1}{N - N_{\text{active}}} \sum_{i \notin A} \theta_i. \qquad (16)$$

Table 1 summarizes expressions (15) and (16) for the cases considered in Figure 1, using $N_{\text{active}} = 4$ and $A = \{1, 30, 35, 85\}$. Note from the table that in the case of a constant activation factor $[f(n) = 0.001]$, more gain from the inactive coefficients is transferred to the active ones, resulting in a better distribution of the adaptation gains.

TABLE 1
Total Gain Distribution of PNLMS
Algorithm over 5000 Iterations

Parameter Values	θ_1	θ_{30}	θ_{35}	θ_{85}	$\theta_{\text{mean}}^{\text{inactive}}$
$\rho = 0.01$	198.0	1868.5	924.5	176.8	19.1
$\rho = 0.05$	80.5	765.3	376.2	72.1	38.6
$\rho = 0.50$	49.9	98.6	49.5	49.5	49.5
$f(n) = 0.001$	283.9	2749.3	1364.0	262.2	3.5

Figure 2 shows the initial behaviour of the active coefficients $w_1(n)$, $w_{30}(n)$, and $w_{35}(n)$, inactive coefficient $w_2(n)$, and gains $g_1(n)$, $g_2(n)$, $g_{30}(n)$, and $g_{35}(n)$. By analyzing Figure 2(a) at the beginning of the learning phase $(0 \le n < 30)$, we observe that $|w_{30}(n)| < |w_1(n)|$, resulting in $g_{30}(n) < g_1(n)$ [see Figure 2(b)]. This is not desirable behaviour, since $g_1(n)$ and $g_{30}(n)$ are not proportional to the magnitude of their corresponding plant coefficients p_1 and p_{30}, respectively, where $|p_1| = 0.1$ and $|p_{30}| = 1.0$, in contrast to the desired condition $g_{30}(n) > g_1(n)$. This fact points out that the PNLMS algorithm may present inadequate behaviour at the beginning of the adaptation process. Such behaviour is considered in the next section for devising a new improved version of the PNLMS algorithm.

3. MODIFIED PNLMS ALGORITHM

Considering the above analysis of the standard PNLMS algorithm, we focus our discussion on the following features:

i) When $w_i(n)$ is an active coefficient, note from (10) that its gain is always proportional to $|w_i(n)|$.

ii) When $w_i(n)$ is inactive, observe from (9) that the gain is not proportional to $|w_i(n)|$.

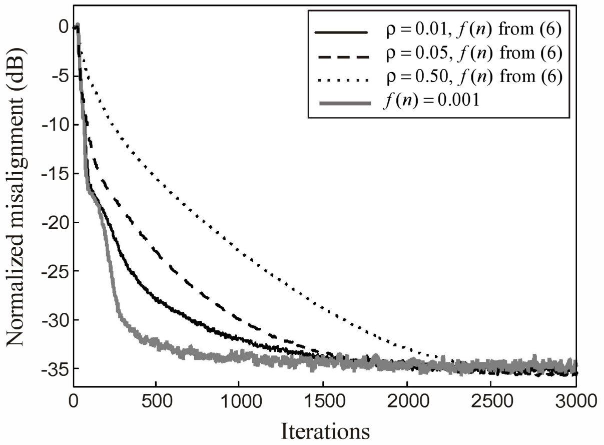
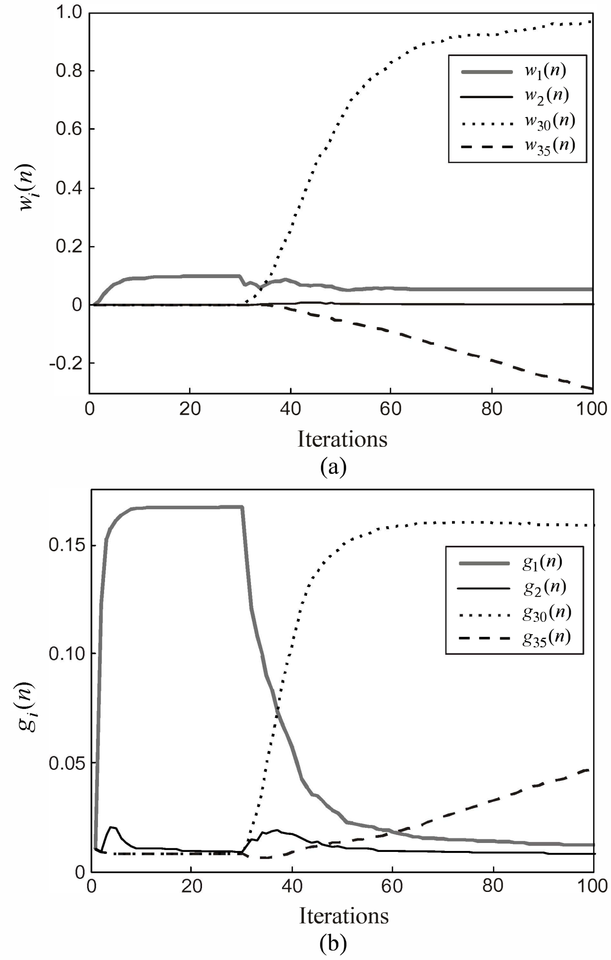


Figure 1 – Normalized misalignment of the PNLMS algorithm for $\mu = 0.5$ and $\delta = 0.01$.



(a)

(b)

Figure 2 – Evolution of PNLMS algorithm variables with $\mu = 0.5$, $\rho = 0.05$, and $\delta = 0.01$. (a) Coefficients $w_1(n)$, $w_2(n)$, $w_{30}(n)$, and $w_{35}(n)$. (b) Gains $g_1(n)$, $g_2(n)$, $g_{30}(n)$, and $g_{35}(n)$.

Feature (i) is desirable since it is in accord with the proportionate philosophy. On the other hand, (ii) is not in accordance with the proportionate concept. Hence, our objective is to overcome this drawback by making the gain $g_i(n)$ tend towards being proportional to $|w_i(n)|$ even when $w_i(n)$ is inactive.

To obtain a truly proportionate algorithm, we associate each gain assigned to an inactive coefficient with an individual activation factor $f_i(n)$ instead of a common one. Thus, replacing $f(n)$ by $f_i(n)$ in (5), the proportionality function is rewritten as

$$\phi_i(n) = \max[f_i(n),\ |w_i(n)|] \tag{17}$$

and accordingly, when $w_i(n)$ is inactive, from (4) and (17), (9) is modified to

$$g_i^{\text{inactive}}(n) = \frac{1}{c(n)} f_i(n). \tag{18}$$

3.1 Conditions Required for the New Activation Factor $f_i(n)$

Now, we establish the required conditions for each individual activation factor $f_i(n)$, which are given as follows:

C1) $f_i(n)$ must converge to the corresponding coefficient magnitude $|w_i(n)|$, i.e.,

$$\lim_{n\to\infty} [f_i(n) - |w_i(n)|] = 0\,, \qquad i = 1, 2, \ldots, N.$$

C2) $f_i(n)$ must always be greater than zero, i.e.,

$$f_i(n) > 0\,, \qquad i = 1, 2, \ldots, N.$$

Thus, if C1 is fulfilled, note from (18) that $g_i^{\text{inactive}}(n)$ tends to be proportional to $|w_i(n)|$ as $n \to \infty$. Moreover, C2 ensures that $g_i(n) > 0$ when $|w_i(n)| = 0$, avoiding the freezing of $w_i(n)$.

3.2 Proposed Approach for Computing $f_i(n)$

Here, our goal is to make the individual activation factor $f_i(n)$ tend towards the magnitude of the corresponding coefficient. Note that at the beginning of the adaptation process $(n = 1)$, the only available data are the estimate of the ith coefficient $w_i(1)$ and the proportionality function $\phi_i(0)$. From (17), one also verifies that $\phi_i(0) = f_i(0)$, since the adaptive filter is started with an initial guess $\mathbf{w}(0) = \mathbf{0}$. Thus, the following formulation is reasonable for computing the activation factor $f_i(1)$:

$$f_i(1) = \gamma |w_i(1)| + (1-\gamma)\phi_i(0) \tag{19}$$

where $0 < \gamma < 1$. Here, the activation factors are initialized with a small positive constant (typically, $f_i(0) = 10^{-2}/N$), such that $f_i(0) > 0$. In (19), the first r.h.s term is due to the intended aim and the second, ensures that $f_i(1)$ be always greater than zero. Now, generalizing this approach for all n, we get

$$f_i(n) = \gamma |w_i(n)| + (1-\gamma)\phi_i(n-1). \tag{20}$$

By considering that no knowledge of the system plant is available *a priori*, it is reasonable to choose $\gamma = 1/2$, weighting equally $|w_i(n)|$ and $\phi_i(n-1)$, thereby obtaining

$$f_i(n) = \frac{1}{2}|w_i(n)| + \frac{1}{2}\phi_i(n-1). \tag{21}$$

By recursion, using (17) and (21) from time 1 to n, one can show that $f_i(n)$ given by (21) fulfils conditions Cl and C2. Note from (21) that now $f_i(n)$ depends on $|w_i(n)|$. So, for proper algorithm operation, it is required that the instantaneous magnitude of the estimated coefficients be proportional to the magnitude of the corresponding plant coefficients. However, $|w_i(n)|$ may not be proportional to $|p_i(n)|$ at the beginning of the adaptation process [see Figure 2(a)], but this can be circumvented by periodically updating $f_i(n)$ only after a learning period of N samples, equal to the adaptive filter length. Therefore, (21) is revised to

$$f_i(n) = \begin{cases} \frac{1}{2}|w_i(n)| + \frac{1}{2}\phi_i(n-1), & n = mN, \quad m = 1, 2, 3, \ldots \\ f_i(n-1), & \text{otherwise}. \end{cases} \tag{22}$$

This expression and (17) characterize the proposed IAF-PNLMS algorithm. Now, each coefficient, either active or inactive, has an associated activation factor $f_i(n)$ computed by (22).

Regarding the computational burden of the proposed algorithm, an additional memory of size $2N$ is required for storing both $\phi_i(n-1)$ and $f_i(n-1)$. On the other hand, the computation of (6) (which is required in the standard PNLMS) is no longer needed for the IAF-PNLMS algorithm, saving N comparisons and one multiplication operation. With respect to the computation of (22), N additions and N multiplications are required every N samples, yielding one addition and one multiplication per iteration.

4. NUMERICAL SIMULATIONS

In this section, MC simulations (average of 100 independent runs) of the NLMS, PNLMS, IPNLMS, and IAF–PNLMS algorithms are carried out for a system identification problem, aiming to compare their convergence speed and response to plant perturbations. In addition, the IAF–PNLMS algorithm gain distribution is assessed. To this end, three examples are presented, considering the same simulation scenario specified in Section 2.2.

4.1 Example 1

In this example, the convergence speed and response to a plant perturbation of the NLMS, PNLMS, IPNLMS, and IAF-PNLMS algorithms are compared. For such, consider that a perturbation in the plant takes place at $n = 2500$, whereby the plant vector $\mathbf{p}$ is changed to $-\mathbf{p}$.

Figure 3(a) illustrates the normalized misalignment curves of the NLMS, PNLMS (with $\rho = 0.05$ and $\delta = 0.01$), IPNLMS, and IAF-PNLMS [with $f_i(0) = 10^{-4}$] algorithms. In the IPNLMS, $\alpha = 0$ (a parameter of that algorithm) is used [6], [7]. By comparing the curves in Figure 3(a), we observe that the IAF-PNLMS algorithm achieves the fastest convergence speed.

4.2 Example 2

In this example, we again compare the convergence speed and response to a plant perturbation of the NLMS, PNLMS, IPNLMS, and IAF-PNLMS algorithms using the same parameter values as in Example 1. Here, at $n = 2500$, the plant vector $\mathbf{p}$ is shifted to the right by 12 samples, changing the position of all active coefficients. In this way, active plant coefficient values equal to $\{0.1,\ 1.0,\ -0.5,\ 0.1\}$, located at positions $\{1,\ 30,\ 35,\ 85\}$, are moved to positions $\{13, 42, 47, 97\}$ after $n = 2500$.

Figure 3(b) shows normalized misalignment curves of the NLMS, PNLMS (with $\rho = 0.05$ and $\delta = 0.01$), IPNLMS (with $\alpha = 0$), and IAF-PNLMS [with $f_i(0) = 10^{-4}$] algorithms. Note from this figure that the IAF-PNLMS algorithm again achieves the fastest convergence.

4.3 Example 3

In this example, we assess the total gain distribution of the IAF-PNLMS algorithm over 5000 iterations. We use the same plant perturbation and conditions as in Example 1. Moreover, to assess the total gain distribution, we first define the ideal activation factor $f_i^{\text{ideal}}(n)$ as

$$f_i^{\text{ideal}}(n) = |p_i| \tag{23}$$

where $|p_i|$ is the magnitude of the ith plant coefficient. This definition is reasonable, since, assuming convergence, $f_i(\infty) \to |w_i(\infty)|$ according to condition C1. Thus, for this simulation scenario, (23) results in

$$f_i^{\text{ideal}}(n) = \begin{cases} 0.1, & i = 1, 85 \\ 1.0, & i = 30 \\ 0.5, & i = 35 \\ 0.0, & \text{otherwise}. \end{cases} \tag{24}$$

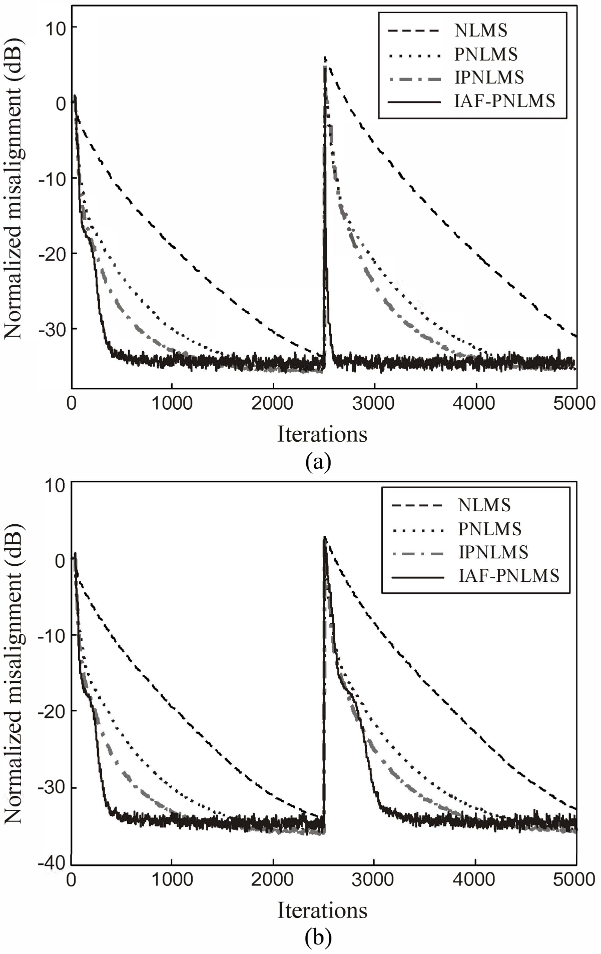


Figure 3 – Normalized misalignment curves using $\mu = 0.5$, $\rho = 0.05$, $\delta = 0.01$, $f_i(0) = 10^{-4}$, and $\alpha = 0$. (a) Example 1: plant sign reversed at $n = 2500$. (b) Example 2: shift of the plant coefficients at $n = 2500$.

Table 2 presents the total gain distribution of the IAF-PNLMS algorithm (over $L = 5000$ iterations) for $f_i(0) = 10^{-4}$. A second total gain distribution for the IAF-PNLMS using $f_i^{\text{ideal}}(n)$ is also shown in this table. Note that the gain distribution obtained for the both cases is numerically very similar, suggesting that the new rule (22) used to obtain $f_i(n)$ is sound. Here, it is important to point out that the use of $f_i^{\text{ideal}}(n)$ is not a practical solution, since in general $|p_i|$ is not known *a priori*. We have introduced this only for comparison purposes.

TABLE 2
Example 3. Total Gain Distribution of
IAF-PNLMS Algorithm over 5000 Iterations

Activation Factors	θ_1	θ_{30}	θ_{35}	θ_{85}	$\theta_{\text{mean}}^{\text{inactive}}$
$f_i(0) = 10^{-4}$	318.2	2860.8	1430.4	282.4	1.1
$f_i^{\text{ideal}}(n)$	300.0	2929.6	1470.8	299.6	0.0

5. CONCLUSIONS

In this work, a new improved version of the PNLMS algorithm, called IAF-PNLMS, is proposed, which uses an individual activation factor for each adaptive filter coefficient. Each activation factor is computed in terms of its corresponding coefficient magnitude, and does not rely on the proportionality and initialization parameters used in the standard PNLMS algorithm. The IAF-PNLMS algorithm provides better gain distribution than the PNLMS and IPNLMS algorithms, leading to an improvement in convergence speed, outperforming both the PNLMS and IPNLMS algorithms for plant impulse responses having high sparseness.

REFERENCES

[1] B. Jelfs, D. P. Mandic, and J. Benesty, “A class of adaptively regularised PNLMS algorithms,” in *Proc. 15th Int. Conf. Digital Signal Process.*, Cardiff, UK, Jul. 2007, pp. 19-22.

[2] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson, Jr., “Exploiting sparsity in adaptive filters,” *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1883-1894, Aug. 2002.

[3] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. New York: Springer-Verlag, 2006.

[4] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancellers,” *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 508-518, Sep. 2000.

[5] S. L. Gay, “An efficient, fast converging adaptive filter for network echo cancellation” in *Proc. Thirty-Second Asilomar Conf. Signals, Systems and Computers*, Monterey, CA, Nov. 1998, vol. 1, pp. 394-398.

[6] J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Orlando, USA, May 2002, vol. 2, pp. 1881-1884.

[7] J. Cui, P. A. Naylor, and D. T. Brown, “An improved IPNLMS algorithm for echo cancellation in packet-switched networks,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Montreal, Canada, May 2004, vol. 4, pp. 141-144.

[8] P. Loganathan, A. W. H Khong, and P. A Naylor, “A sparseness controlled proportionate algorithm for acoustic echo cancellation,” in *Proc. European Signal Process. Conf. (EUSIPCO 2008)*, Lausanne, Switzerland, Aug. 2008.

[9] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *J. Mach. Learning Research*, vol. 5, pp. 1457-1469, Nov. 2004.