Adaptive Combination of Proportionate Filters for Sparse Echo Cancellation

Jerónimo Arenas-García*, *Member, IEEE*, Aníbal R. Figueiras-Vidal, *Senior Member, IEEE*

Abstract

Proportionate adaptive filters, such as those based on the improved proportionate normalized least-mean-square (IPNLMS) algorithm, have been proposed for echo cancellation as an interesting alternative to the normalized least-mean-square (NLMS) filter. Proportionate schemes offer improved performance when the echo path is sparse, but are still subject to some compromises regarding their convergence properties and steady-state error. In this paper, we study how combination schemes, where the outputs of two independent adaptive filters are adaptively mixed together, can be used to increase IPNLMS robustness to channels with different degrees of sparsity, as well as to alleviate the rate of convergence vs steady-state misadjustment tradeoff imposed by the selection of the step size. We also introduce a new block-based combination scheme which is specifically designed to further exploit the characteristics of the IPNLMS filter. The advantages of these combined filters are justified theoretically and illustrated in several echo cancellation scenarios.

Index Terms

Combination filters, proportionate filters, echo cancellation, sparse channel identification

EDICS: AUD-ECHO Echo Cancellation

The authors are with the Dep. of Signal Theory and Communications, Universidad Carlos III de Madrid, 28911 Leganés, Spain. (e-mail:{jarenas,arfv}@tsc.uc3m.es). This work has been partly supported by MEC project TEC2005-00992 and CAM project S-0505/TIC/0223.

I. INTRODUCTION

Adaptive echo cancellation, both acoustic and electrical, is a key component of modern communication networks. The overall echo cancellation process is illustrated in Fig. 1. The echo is produced when the far-end signal activates a (possibly time-varying) echo path, $\mathbf{w}_\mathrm{o}(n)$. This echo signal is superimposed upon the near-end signal, $s(n)$, which is possibly contaminated by additive noise $e_0(n)$. The goal of an echo canceler is to produce a replica $y(n)$ of the echo signal which can be used to remove the echo before the signal is delivered to the far-end.

In this paper, we are interested in modeling sparse or quasi-sparse echo channels, in which only a small fraction of the weights of the impulse response $\mathbf{w}_\mathrm{o}(n)$ are significantly different from zero (the so-called active coefficients). Such echo paths are typically encountered in acoustic and network echo cancellation [1], [2], including also internet telephony [3], [4], where echo-path impulse responses are of short duration, but present unknown delays. Therefore, it becomes necessary to use echo cancelers with a long memory, which can be implemented with adaptive filters with hundreds or even thousands of weights, of which only a few will significantly differ from zero after convergence. Following [4], we define the degree of sparsity of a channel as a qualitative measure ranging from strongly dispersive (when most of the coefficients of $\mathbf{w}_\mathrm{o}(n)$ are active) to strongly sparse. The degree of sparsity of a particular echo path is typically not known beforehand, and it can even be time-varying. For this reason, it is desirable to develop schemes that show robust performance to different levels of sparseness in the echo channel.

It is a well-known fact that adaptive schemes which distribute the adaptation energy equally among all filter coefficients, such as least-mean-square (LMS) and normalized LMS (NLMS), exhibit a very slow convergence for filters with many taps [5], [6], what makes the application of such schemes unpractical for sparse echo cancellation. To alleviate this problem in acoustic echo cancellation applications, Makino et al. [7] introduced the exponentially-weighted step size NLMS (ES-NLMS), which assigns a different adaptation speed to each coefficient of the echo canceler:

$$w_m(n+1) = w_m(n) + \mu_m \frac{e(n)}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)} x_m(n), \quad m = 1, \ldots, M, \tag{1}$$

where M is the filter length, $w_m(n)$ and $x_m(n)$ are the mth components of the filter weights and the input regressor at time n, $\mathbf{w}(n)$ and $\mathbf{x}(n)$, respectively, and $e(n) = d(n) - y(n)$ is the error incurred by the filter, $d(n)$ being the desired response and $y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$ the filter output. Parameter δ is a small constant which prevents division by 0. In the ES-NLMS algorithm, step sizes $\{\mu_m\}_{m=1}^M$ decay exponentially with m, following the typical profile of most acoustic echo paths, so that the most active coefficients receive

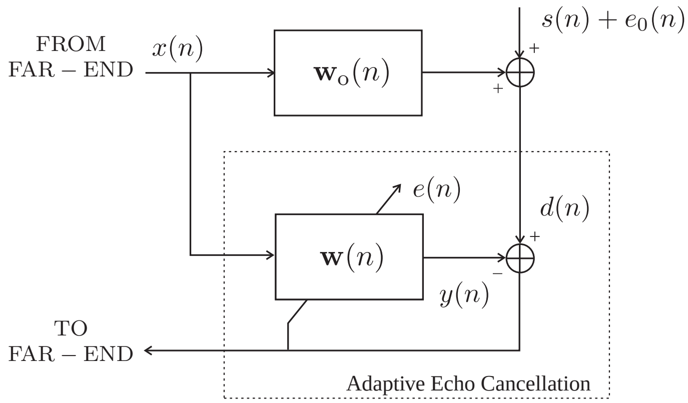


Fig. 1. Block diagram for an adaptive echo cancellation configuration.

faster adaptation. The effective application of this adaptive scheme requires, therefore, some knowledge regarding the bulk delay and weight-decay constant, which is normally not available.

Inspired by the same working principles, the proportionate NLMS algorithm (PNLMS) [8] makes the adaptation step for each tap proportional to the current absolute value of the estimated weight, i.e,

$$w_m(n+1) = w_m(n) + \mu_m(n) \frac{e(n)}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)} x_m(n), \quad m = 1, \ldots, M, \tag{2}$$

where $\mu_m(n) \propto |w_m(n)|$. Therefore, the PNLMS algorithm tries to accelerate the convergence of the filter by adapting faster the weights corresponding to the active region of the sparse echo path. The advantage of PNLMS over ES-NLMS is that it does not assume any other *a priori* knowledge about the echo channel but its sparsity.

Following the introduction of the PNLMS filter, several other approaches have focused on improving different aspects of this scheme. On the one hand, it is found that, after a very fast initial response, PNLMS convergence slows down, something that can be corrected by a more adequate selection of the adaptation energy for each tap [9], [10]. On the other hand, PNLMS behavior degrades significantly when identifying not-so-sparse echo channels. To obtain schemes robust to the presence of channels with different degrees of sparseness, one can switch between PNLMS and NLMS [11], [12].

Another scheme which tries to improve the robustness of PNLMS to dispersive channels is the so-called improved PNLMS algorithm (IPNLMS) [13], to which we will devote our attention in this paper. The coefficients of an IPNLMS filter are adapted according to

$$w_m(n+1) = w_m(n) + \mu_m(n) e(n) x_m(n), \quad m = 1, \ldots, M, \tag{3}$$

$$\mu_m(n) = \frac{\mu \, g_m(n)}{\delta + \sum_{k=1}^{M} g_k(n) x_k^2(n)}, \tag{4}$$

where μ is the step size of the filter. The gain for each weight, $g_m(n)$, is calculated using

$$g_m(n) = (1-\kappa)\frac{1}{2M} + (1+\kappa)\frac{|w_m(n)|}{\epsilon + 2\|\mathbf{w}(n)\|_1}, \quad (5)$$

where ϵ is a small positive constant, $\|\mathbf{w}(n)\|_1 = \sum_k |w_k(n)|$, and κ is a constant between -1 and 1, which establishes a tradeoff between the standard NLMS filter ($\kappa = -1$) and basic PNLMS ($\kappa = 1$).

From (4) and (5), it is evident that the step size associated to each coefficient increases with the absolute value of that coefficient. Consequently, like in the PNLMS case, IPNLMS spends more energy adapting the active coefficients, thus converging faster than NLMS.

For completeness, we should also mention here the EG$\pm$ algorithm [14] as another filter which, by using exponentiated gradient adaptation, weights unevenly the adaptation of the different taps. Nevertheless, it has been theoretically justified [15], [4] that IPNLMS is a very good approximation of the EG$\pm$ filter, while being more convenient from a practical point of view.

In spite of IPNLMS generally achieving faster convergence than NLMS, the selection of its parameters μ and κ must consider the following compromises:

1) As any other gradient-based adaptive filter, IPNLMS is subject to a speed vs precision tradeoff, i.e., a large step size results in faster convergence, while the residual misadjustment is reduced for small μ.
2) Parameter κ imposes a behavior tradeoff for channels with different degrees of sparsity [13]. For strongly sparse channels the fastest convergence is obtained by the PNLMS filter ($\kappa = 1$), but such a selection leads to suboptimal performance for not-so-sparse channels. Therefore, the best value of κ for a given scenario depends on the actual degree of sparsity of the echo channel.

The adaptive combination of adaptive filters has proved to be an effective approach to improve the performance of adaptive schemes and to simplify their use. The basic idea of the algorithms in [16], [17] is to combine adaptive filters with different settings, so that the combination behaves, at each iteration, as the best component filter (or even better than any of them [18]). Since their introduction, these schemes have been used in several areas of adaptive signal processing, including blind equalization [19], and signal characterization [20], among others.

Although the combination scheme of [16] was originally proposed to improve the speed vs precision tradeoff, in [21] we presented some preliminary experimental results showing that it can also be applied to very efficiently increase the robustness of IPNLMS filters to channels with different degrees of sparsity. In this paper, we further study the performance of convex combinations of two IPNLMS filters, extending our previous contribution [21] in several ways. We introduce a novel block-based combination scheme which is specifically designed to get the most out of adaptive filters with proportionate adaptation. We present also a performance analysis of our echo cancellation schemes, showing that the combination of filters incurs in a lower steady-state error than any of its IPNLMS components.

The overall structure of the paper is as follows: Next section briefly reviews the basic algorithm of [16] for combining two adaptive filters, and explains how it can be useful to boost the performance of IPNLMS filters. Section III introduces the novel block-based combination scheme for adaptive filters. Experimental evidence on the advantages of both the basic and block-based combination schemes is given in Section IV, while some theoretical insight about their working principles is provided in the Appendix. Section V summarizes the main conclusions of this work.

II. Adaptive Combination of IPNLMS Filters

The configuration of the adaptive combination scheme of two adaptive filters presented in [16] is illustrated in Fig. 2. To get a good performance from the combination, both component filters, $\mathbf{w}_1$ and $\mathbf{w}_2$, are independently adapted, using their own rules and settings. The overall filter output, $y(n)$, is calculated as a convex combination of the outputs of the two component filters, $y_i(n),\ i = 1, 2$:

$$y(n) = \lambda(n)y_1(n) + [1 - \lambda(n)]y_2(n), \tag{6}$$

where $\lambda(n)$ is a mixing parameter which is kept in the range $[0, 1]$ by defining it as the output of a sigmoid activation function,

$$\lambda(n) = \text{sgm}[a(n)] = \frac{1}{1 + e^{-a(n)}}. \tag{7}$$

At each iteration step, $a(n)$ will be adapted as described below, and then $\lambda(n)$ will be obtained from (7). Since the relation between $a(n)$ and $\lambda(n)$ is one to one, we will indistinctly refer to them as mixing parameters.

If the mixing parameter is appropriately adapted, the combination will keep the best characteristics of each component. Following [16], $a(n)$ can be adapted to minimize the square error of the overall filter according to the following stochastic gradient descent rule:

$$\begin{aligned} a(n+1) &= a(n) - \mu_a \frac{\partial e^2(n)}{\partial a(n)} \\ &= a(n) + \mu_a e(n)[y_1(n) - y_2(n)]\lambda(n)[1 - \lambda(n)], \end{aligned} \tag{8}$$

μ_a being a step-size parameter for the adaptation of $a(n)$. For practical reasons, $a(n)$ is kept within $[-4, 4]$, so that its adaptation does not stop because of factor $\lambda(n)[1 - \lambda(n)]$ in (8) being too close to 0 (for more details on this truncation procedure, please refer to [18]).

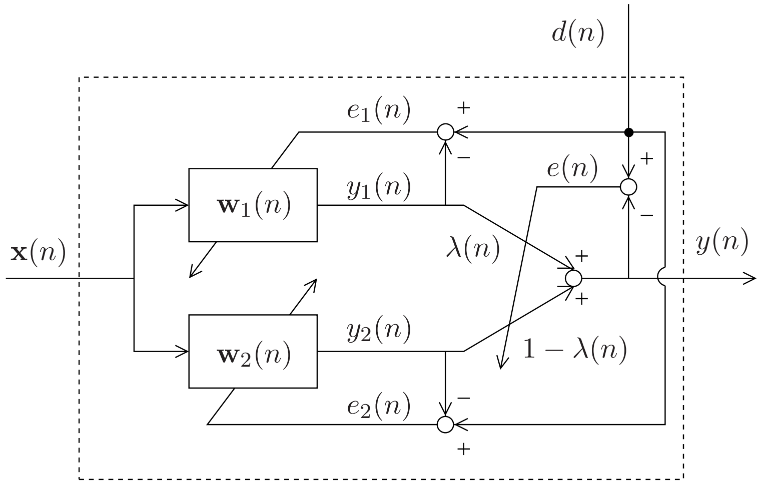


Fig. 2. Adaptive combination of two adaptive filters. Each component is adapted according to its own rules and error, while the mixing parameter, $\lambda(n)$, is updated to minimize the overall error.

The steady-state performance of the combination scheme we have just reviewed has been theoretically analyzed in [18], showing that it enjoys *universal capabilities* with respect to the component filters. In other words, the combination performs at least as well as the best component filter and, under certain conditions, better than any of them (and, in this case, we say that the scheme performance is *better-than-universal*). Note that this result holds independently of the kind of adaptive filters used for the combination. The *better-than-universal* behavior can be explained from a reduction in the variance of the error of the overall filter, provided that the cross-correlation between the component filters errors is sufficiently small [18]. In Section IV and in the Appendix we will show that such a situation is observed for the particular case of a combination of IPNLMS adaptive filters.

The adaptive combination scheme can be used to improve the convergence rate vs misadjustment trade-off of IPNLMS filters, as well as their robustness to echo paths with different degrees of sparsity. To do so, we propose to combine two IPNLMS filters, selecting their parameters, $\{\mu_1, \kappa_1\}$ and $\{\mu_2, \kappa_2\}$, in one of the following ways:

a) $\mu_1 > \mu_2$ and $\kappa_1 = \kappa_2$: This configuration is intended to keep the faster convergence of the filter with step size μ_1, while achieving the lower steady-state error of the filter with small step size; thus the overall convergence rate vs steady-state performance tradeoff is improved.

b) $\mu_1 = \mu_2$, $\kappa_1 < 0$, $\kappa_2 \approx 1$: With this selection, the combined filter shows improved convergence and robustness against channels with different degrees of sparsity, i.e., the combination retains the faster initial convergence of PNLMS for very sparse channels[1], while behaving as an IPNLMS with κ_1 in a later stage of the adaptation, when PNLMS convergence is known to slow-down. The filter with κ_1 provides also robustness against channels with a low degree of sparsity.

Switching between the PNLMS and the NLMS algorithms, which can be seen close in spirit to the second configuration proposed above, has already been explored in [11], [12]. The present combination of IPNLMS filters can be seen as a more flexible approach in the sense that, rather than using a hard commutation between both adaptation schemes, it uses a soft combination of two component filters which are adapted in parallel, and which can be different in any form. We will see in the experiments section that, by doing so, not only a faster convergence is achieved, but in many situations it is also possible to get a lower identification error than from any of the components alone.

III. Block-based Combination Scheme

The combination scheme we have just presented can alternatively be considered as a filter with weights

$$\mathbf{w}(n) = \lambda(n)\mathbf{w}_1(n) + [1 - \lambda(n)]\mathbf{w}_2(n), \tag{9}$$

so that the same mixing parameter is used for all weights. However, when combining IPNLMS filters it may be preferable to use a different mixing parameter for each coefficient.

To illustrate this, let us consider the combination of an NLMS and a PNLMS filters (i.e., $\kappa_1 = -1$ and $\kappa_2 = 1$). In the Appendix, we will carry out a steady-state analysis of the IPNLMS algorithm, where we show that the contribution of each tap-weight to the steady-state misalignment of the filter is approximately proportional to

$$\Psi_m(\infty) = \frac{\bar{g}_m}{2 - \mu\bar{g}_m}, \tag{10}$$

which is an increasing function of the adaptation gains corresponding to the optimum weight vector [see (5)], i.e.,

$$\bar{g}_m = (1-\kappa)\frac{1}{2M} + (1+\kappa)\frac{|w_{\mathrm{o},m}|}{2\|\mathbf{w}_\mathrm{o}\|_1}, \tag{11}$$

where $w_{\mathrm{o},m}$ is the mth component of $\mathbf{w}_\mathrm{o}$, and where we are assumming a vanishingly small ϵ.

Figure 3 represents $\Psi_m(\infty)$ for the different coefficients of an NLMS and a PNLMS filters, when both algorithms are used to identify the echo-path impulse response of a hybrid sampled at a rate of 8 kHz. The length and step size used for both filters are $M = 512$ and $\mu = 0.5$, respectively. From Fig. 3 it can be seen that the contribution of filters weights to the residual gradient noise is very different for both

[1]Note that, for $\kappa_2 \approx 1$, the second filter behaves approximately as a PNLMS filter [13].

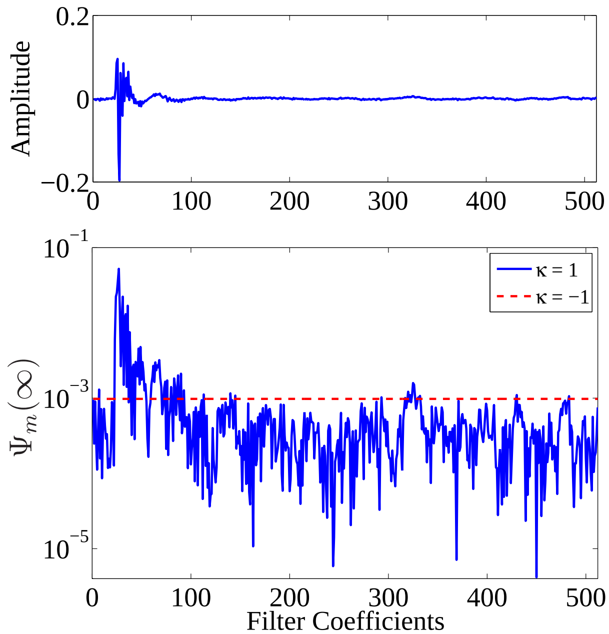


Fig. 3. Contribution to steady-state gradient noise of the different coefficients of an NLMS and a PNLMS filters ($\kappa = -1$ and $\kappa = 1$, respectively) with $M = 512$ and $\mu = 0.5$. The upper panel represents the impulse response of the echo path identified by both adaptive schemes.

algorithms. For the NLMS filter, all taps are assigned the same adaptation gain [$g_m(n) = 1/M, \ \forall m$] and, consequently, they contribute evenly to the steady-state error of the filter. On the contrary, active coefficients concentrate most of the adaptation energy of the PNLMS filter, therefore, the estimation provided by PNLMS of such active weights is affected by a large amount of gradient noise (note the log scale in the vertical axis) [4].

In the light of this result, we can conclude that a filter that combines the estimation of active coefficients from the NLMS filter with that of non-active ones provided by PNLMS, would incur in a lower steady-state misadjustment than any of the original filters.

This idea has already been exploited in [3], [4], where the authors proposed a modification of the IPNLMS algorithm that explicitly classifies the filter weights into active and non-active using a threshold value. Here, we propose to extend the principles of the adaptive convex combination approach, and mix together the coefficients of the two component filters using different mixing coefficients for the different weights. In contrast to the improved IPNLMS (IIPNLMS) algorithm of [4], our approach does not require to explicitly distinguish between active and non-active coefficients, but instead relies on the minimization of the quadratic error to decide how to better combine every pair of coefficients, allowing soft mixtures, if necessary.

Rather than using a different mixing parameter for each weight, as it was done in [17], the local structure of typical echo channels suggests using only one parameter for each group of adjacent coefficients. By doing so, we can significantly reduce the computational complexity of the combination. Then, if we consider L blocks, each one consisting of $M_b = M/L$ coefficients, the output of the block-based combination scheme will be:

$$y(n) = \sum_{l=1}^{L} \sum_{m \in \mathcal{I}_l} \lambda_l(n) w_{m,1}(n) x_m(n) + [1 - \lambda_l(n)] w_{m,2}(n) x_m(n), \tag{12}$$

where $\lambda_l(n)$ is the mixing parameter associated to the lth block, $w_{m,1}(n)$ and $w_{m,2}(n)$ refer to the mth taps of each of the component filters weights, and $\mathcal{I}_l = [(l-1)M_b + 1, \ldots, l \cdot M_b]$ indexes the coefficients belonging to the lth block.

Adaptation of the L mixing parameters can be carried out in a similar way to the adaptation of the mixing parameter of the standard combination. If we define $\lambda_l(n) = \text{sgm}[a_l(n)],\ l = 1, \ldots, L$, the minimization of the overall square error using stochastic gradient descent leads to:

$$\begin{aligned} a_l(n+1) &= a_l(n) - \mu_a \frac{\partial e^2(n)}{\partial a_l(n)} \\ &= a_l(n) + \mu_a e(n) \lambda_l(n) [1 - \lambda_l(n)] \sum_{m \in \mathcal{I}_l} [w_{m,1}(n) - w_{m,2}(n)] x_m(n), \end{aligned} \tag{13}$$

for $l = 1, \ldots, L$, where, again, $a_l(n)$ should be restricted to interval $[-4, 4]$.

The computational complexity of the proposed combination is roughly twice that of the basic IPNLMS (for the two IPNLMS filters running in parallel). The additional number of multiplications that are required for calculating the overall output and adapting the mixing parameters is $6L$ (with $L \ll M$ normally, and $L = 1$ for the scheme of Section II), which is usually much smaller than the number of products required by the adaptation of the components. If necessary, the computational cost could be significantly reduced introducing selective-tap updates for the component filters [4], [22]-[24].

IV. Experiments

In this section, we evaluate the ability of both the standard and the block-based combination schemes to improve the performance of IPNLMS using echo cancellation scenarios similar to those encountered in, e.g., [3], [4], [12], [13]. Different echo paths will be used, all with length $M = 512$ and an attenuation of 10 dB. The far-end signal, $x(n)$, from which input regressors are taken as $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-M+1)]^T$, is random noise with zero mean and variance 1. For our experiments, we will consider both white Gaussian noise (WGN) input, and a case in which $x(n)$ is USASI noise with a speech-like spectrum [25][2]. The output added noise, $e_0(n)$, is also a white Gaussian noise, whose power is set to get an SNR of 20 dB in the reference signal. We assume that no near-end signal is present (i.e., $s(n) = 0$), since the adaptation of echo cancelers is usually halted when double-talk situations are detected [26].

Adaptive filters weights have been initialized with the zero vector. Parameter settings for the IPNLMS component filters are $\epsilon = 10^{-6}$ and $\delta = 0$, while different μ and κ values are used to illustrate the different benefits of the combination. The initial value of the mixing parameter is $a(0) = 0$ (for which $\lambda(0) = 0.5$), while the step size for its adaptation has been fixed to $\mu_a = 100$ in all cases.

For comparison purposes, we will consider the excess mean-square error (EMSE), $\text{EMSE}(n) = E\{[e(n) - e_0(n)]^2\}$, estimated using an average of 1000 runs of the algorithms.

A. Improving the convergence speed vs steady-state misadjustment tradeoff

We illustrate here how the convex combination scheme can be exploited, in a very easy and efficient way, to improve the speed of convergence vs steady-state performance compromise of IPNLMS filters. The problem of controlling the learning rate of echo cancelers has also been addressed in [27], [28].

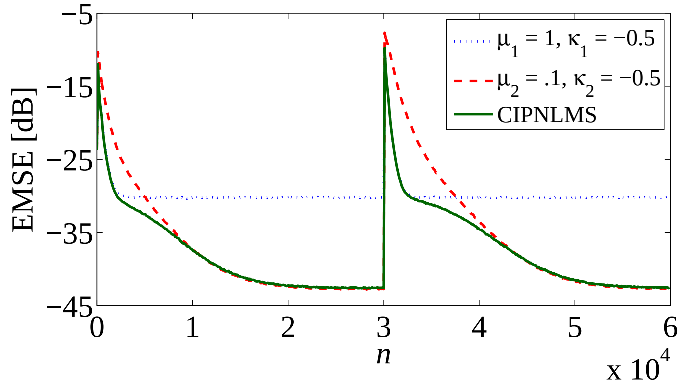
For this first set of experiments we have combined two IPNLMS filters with $\kappa_1 = \kappa_2 = -0.5$, as recommended in [13], using a different step size for each component: $\mu_1 = 1$ and $\mu_2 = 0.1$. The real echo path used in this subsection is the channel whose impulse response was depicted in Fig. 3, upper panel[3]. To study the ability of the algorithm to reconverge, a change in the echo path is simulated after a number of steps (see Fig. 4) by circularly shifting all coefficients 50 positions to the right.

Fig. 4(a) illustrates filter performance for WGN input. EMSE evolution is displayed for both component filters, as well as for their combination using the algorithm in Section II, to which we will refer hereafter as combined IPNLMS (CIPNLMS). We can see that the μ_1 IPNLMS filter shows faster convergence, both initially and after the change in the echo path. The filter with μ_2 presents slower convergence, but it achieves a steady-state EMSE more than 10 dB smaller. As expected, the CIPNLMS filter is able to put together fast convergence and reduced steady-state misadjustment. In addition to this, we can check that CIPNLMS is able to simultaneously outperform both components at some iterations during the convergence.

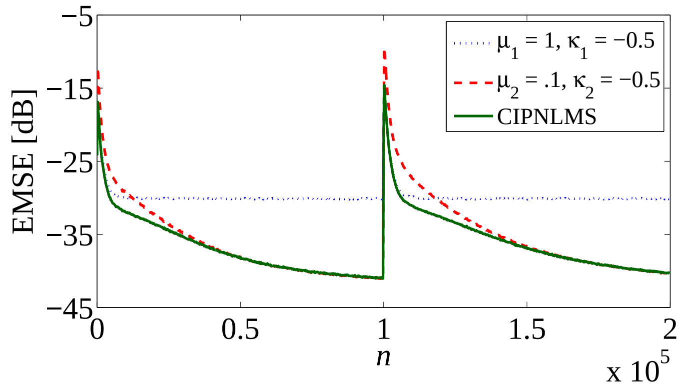
In Fig. 4(b) we illustrate the performance achieved for $x(n)$ being USASI noise. The presence of a

[2]USASI noise was generated using the VOICEBOX toolbox for MATLAB (`http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`).

[3]This echo channel has been previously used in [13], [26].



(a)



(b)

Fig. 4. Mean-square performance of two IPNLMS filters with different step sizes, and of their adaptive combination (CIPNLMS). A change in the real echo path is simulated after 30000 and 100000 steps (for the upper and lower subfigures) by circularly shifting all coefficients 50 positions to the right. (a) $x(n)$ is white Gaussian noise. (b) $x(n)$ is USASI noise with a speech-like spectrum.

colored input causes a slower convergence of the component IPNLMS filters (note the scaling of the time axis) and, therefore, of their combination. Apart from this, the results shown in Fig. 4(b) can be discussed in very similar terms to the case of white Gaussian input.

B. Improving the robustness to channels with different degrees of sparsity

Next, we would like to illustrate how the CIPNLMS algorithm can be exploited to improve the robustness of IPNLMS to channels with different degrees of sparsity, even when the number of active coefficients changes during the simulation. To do so, we have generated a synthetic channel with 256 active coefficients taken from a random Gaussian distribution, and scaled to introduce 10 dB channel attenuation. In the second part of the simulation the echo path commutes to a different channel with only

16 active coefficients. These two artificial echo paths, which have been represented in Fig. 5, will be referred as the dispersive and the strongly sparse echo channels, respectively.

As it was explained at the end of Section II, the adaptive combination approach can be used to improve IPNLMS robustness to channels with different degrees of sparsity by selecting $\mu_1 = \mu_2$, $\kappa_1 < 0$, and $\kappa_2 \approx 1$. Consequently, in this subsection, step sizes are adjusted to $\mu_1 = \mu_2 = 0.5$; as for parameter κ, we have selected $\kappa_1 = -0.5$ and $\kappa_2 = 0.9$, so that the first filter provides good convergence properties when the echo path is not very sparse, while the second component behaves similarly to a PNLMS filter, thus improving convergence for very sparse channels. These are reasonable settings when the degree of sparsity is not known beforehand.

Fig. 6 shows CIPNLMS performance both for WGN and USASI inputs, and illustrate that the combination scheme keeps again the best characteristics of each component. For white $x(n)$ (Fig. 6(a)), the second (PNLMS) component fails at achieving a right convergence during the first part of the simulation, when the channel is not very sparse. However, CIPNLMS remains robust to this situation by following the first filter. After the changes at $n = 15000$ and $n = 50000$ (for the white and colored input signals, respectively), the echo path becomes very sparse, and the second component exhibits a very fast convergence, which is also inherited by the combination. Then, we can conclude that the combination filter is robust to channels with different degrees of sparsity, in the sense that it achieves the very fast convergence of PNLMS if possible (i.e., for very sparse echo paths), while remaining robust to channels with wider active regions.

It is also interesting to notice that the steady-state EMSE of CIPNLMS can be simultaneusly smaller than those of both components, something which is specially clear during the last iterations in Fig. 6(a). This *better-than-universal* steady-state behavior is a side advantage of the combination approach, and will be theoretically justified with the analysis carried out in the Appendix.

For comparison purposes, we have also studied the behavior in this scenario of the sparseness-controlled IPNLMS (SC-IPNLMS) filter of [29]. This algorithm is based on a sparseness measure [29, Eq. (12)] which is exploited to directly control the contribution of the NLMS and proportionate terms to the gain factors of an IPLMS scheme (see (5)). SC-IPNLMS parameter settings are $\mu_{\text{SC}} = 0.5$ and $\alpha = -0.5$. The adjustment for α is slightly different from that used in [29] ($\alpha = -0.75$). Using $\alpha = -0.5$ allows a more appropriate comparison with CIPNLMS (this parameter is equivalent to κ in this paper) and, in fact, provides some minor performance gains in the scenarios we have studied.

Fig. 7 compares the performance of the CIPNLMS and SC-IPNLMS schemes. During the first part

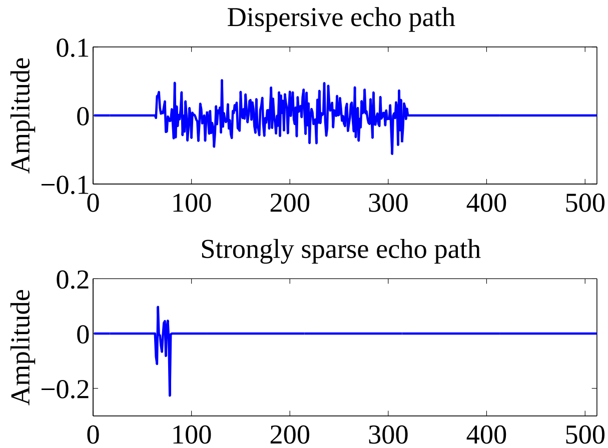


Fig. 5. Impulse response of two artificially generated echo channels, the first being a dispersive channel with 256 active coefficients, and the second being strongly sparse, with only 16 active coefficients.

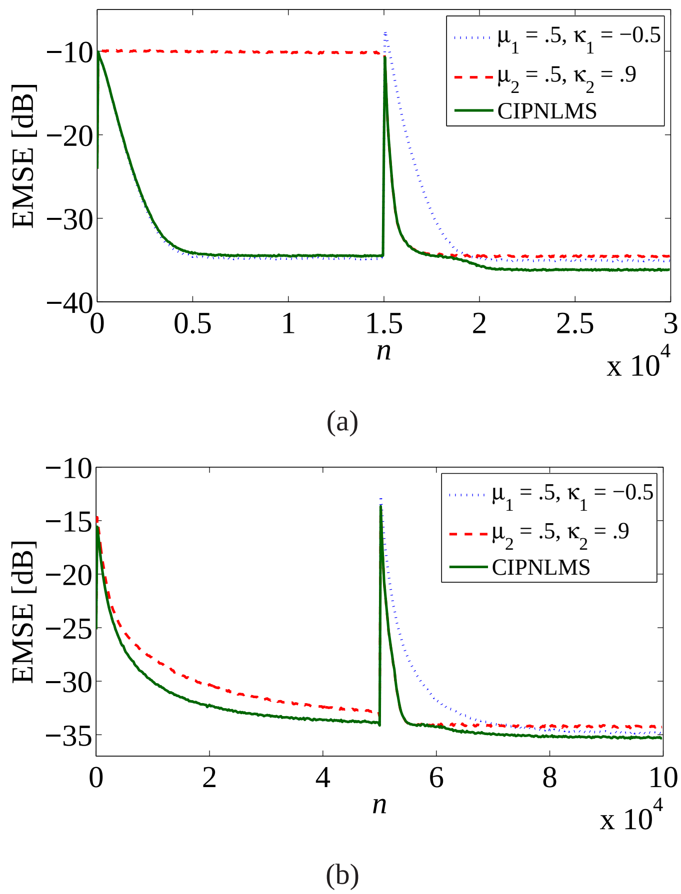
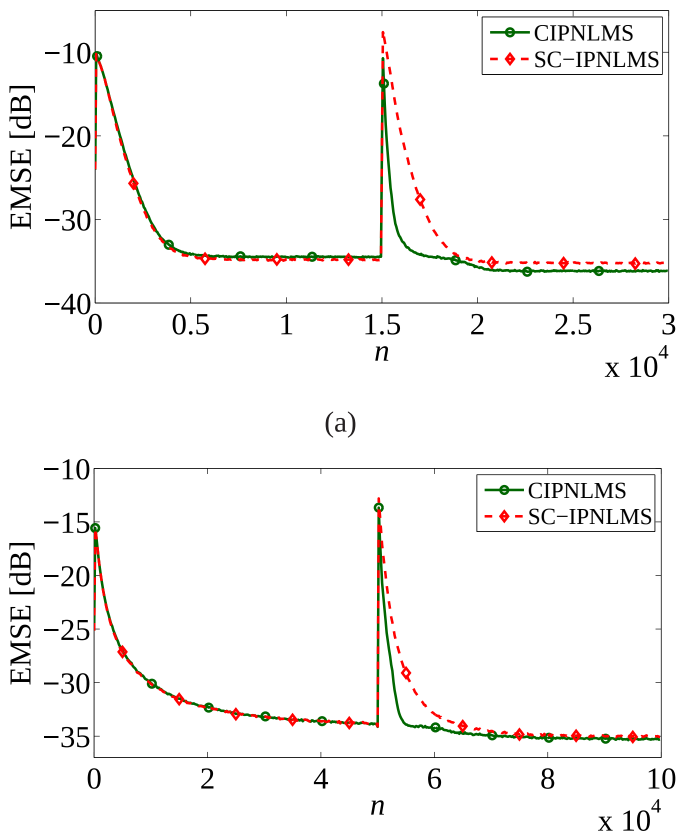


Fig. 6. Performance of the combination scheme when combining IPNLMS filters with the same step size and different κ. The echo path has initially 256 active weights, while only 16 non-zero coefficients are present after the plant change. (a) $x(n)$ is WGN. (b) $x(n)$ is USASI noise.



(b)

Fig. 7. EMSE evolution for CIPNLMS ($\mu_1 = \mu_2 = 0.5$, $\kappa_1 = -0.5$, $\kappa_2 = 0.9$, and $\mu_a = 100$) and SC-IPNLMS (algorithm from [29], $\mu_{\text{SC}} = 0.5$ and $\alpha = -0.5$). The echo path has initially 256 active weights, while only 16 non-zero coefficients are present after the plant change. (a) $x(n)$ is WGN. (b) $x(n)$ is USASI noise.

of the experiment, when the echo path has 256 active coefficients, both algorithms show very similar behaviors, remaining robust to this dispersive channel. Following the change in the echo channel, however, CIPNLMS exhibits a faster convergence than SC-IPNLMS, both for white and colored input signals, Thus, we can conclude that CIPNLMS is able to better exploit the advantages of proportionate adaptation in the presence of strongly sparse echo paths, a behavior inherited from its second component (with $\kappa_2 = 0.9$).

C. Hierarchical combination of IPNLMS filters

In the two previous subsections we have illustrated that the adaptive combination of two adaptive filters can be used to improve two different aspects of the IPNLMS algorithm; now, we explain how the combination approach can be exploited to build an iterated architecture with the goal of simultaneously

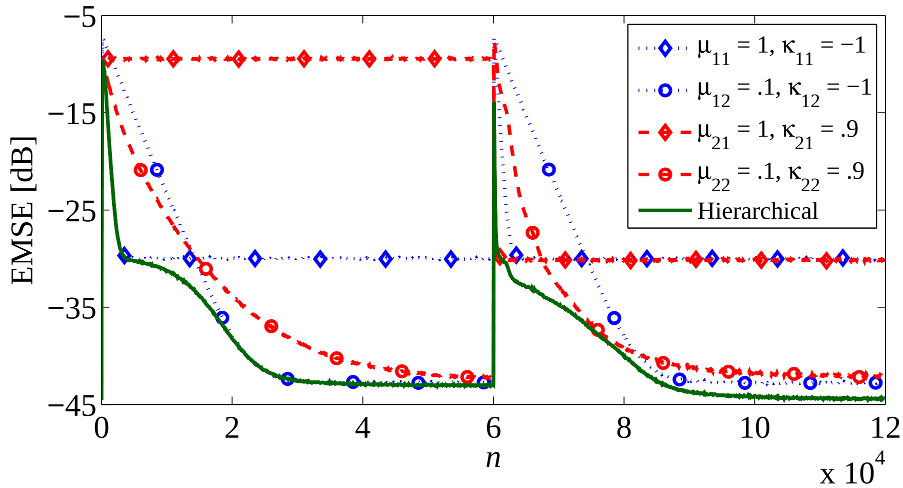


Fig. 8. EMSE evolution for WGN input of a hierarchical combination with four IPNLMS filters with different settings. The echo path has initially 256 active weights, while only 16 non-zero coefficients are present after the change.

improving the convergence vs residual error tradeoff and the robustness to channels with different degrees of sparseness. The basic idea of the proposed hierarchical two-layer combination is to combine four IPNLMS filters. In the first level, we combine filters with the same κ and different step sizes, whereas in the second level we combine the outputs of the filters resulting from first-level combinations.

EMSE evolution for such a hierarchical scheme has been depicted in Fig. 8 for white input. As before, the dispersive echo channel is used initially, and an abrupt change is simulated by commuting to the strongly sparse channel at step 60000. Parameter settings for the four component filters are shown in the legend of the figure. During the first part of the experiment, IPNLMS components with $\kappa = 0.9$ (thus, behaving like PNLMS filters) perform poorly, and the hierarchical scheme behaves like a convex combination of two NLMS filters with large and small step sizes. In the second half of the simulation, when the echo path is strongly sparse, the situation reverses: PNLMS-like components are the best performing among the four component filters, and the hierarchical scheme inherits their superior convergence.

Obviously, the cost we pay for using four component filters in parallel is an increased computational complexity, which is roughly twice that of the CIPNLMS scheme. As noted in Section III, the computation requirements of these schemes can be reduced by using selective-tap updates for the component filters.

D. Block-based combination of IPNLMS filters

Next, we illustrate the performance of the block-based combination of IPNLMS filters (B-CIPNLMS) presented in Section III. As already explained, apart from offering robustness to channels with different degrees of sparsity, this combination scheme is specially designed to reduce the overall EMSE when the combined IPNLMS filters have different values of κ. Therefore, we have selected $\mu_1 = \mu_2 = 0.5$, $\kappa_1 = -1$ and $\kappa_2 = 0.9$, so that we can roughly interpret the combination as a mixture of an NLMS and a PNLMS filters. Simulations have been carried out using $L = 16$ blocks of $M_b = 32$ coefficients each.

Simulation results using the real echo channel (see Fig. 3, upper panel) are presented in Fig. 9 for WGN input. As in Subsection IV-A, an abrupt change in the echo path is simulated by circularly shifting all coefficients 50 positions to the right. In the upper panel the evolutions of the EMSEs of both the component filters and the overall combination are depicted, showing that the block-based combination achieves a better convergence than any of the components, while reducing, as expected, their steady-state EMSE.

This steady-state EMSE decrement can be explained from the evolution of some of the mixing parameters shown in the lower panel of Fig. 9. For instance, $\lambda_2(n)$ is associated to a block which includes some active coefficients during the first part of the simulation. As a consequence, it initially gets close to 0, following the fast convergence of PNLMS, and, after a while, it increases towards 1, to benefit from the smaller gradient noise achieved by the NLMS component for the coefficients in that block. After the change in the channel, all coefficients in the second block become inactive, which explains why $\lambda_2(n)$ goes down to zero during the second half of the experiment. Similarly, from the evolution of $\lambda_4(n)$, we can see that the fourth block corresponds to a set of coefficients which only become active after the abrupt change in the channel.

Since B-CIPNLMS is designed to reduce the steady-state EMSE of the combination, it is worth studying the influence of the block-size M_b on such error. Thus, we have repeated the previous experiment for different values of M_b. Steady-state EMSEs have been estimated by averaging the filter error during 20000 iterations after a complete convergence of the algorithm. According to the results displayed in Fig. 10, B-CIPNLMS is not very sensitive to the selection of this parameter, and reductions of about 2 dB with respect to the steady-state EMSE of the component filters are already obtained for $M_b = 128$. Due to the local structure of the echo path, using smaller blocks does not offer significant additional EMSE reductions. On the contrary, reducing the block-size results in an increment of the number of blocks ($L = 512/M_b$) and, therefore, of the computational cost of the algorithm. It is worth mentioning that

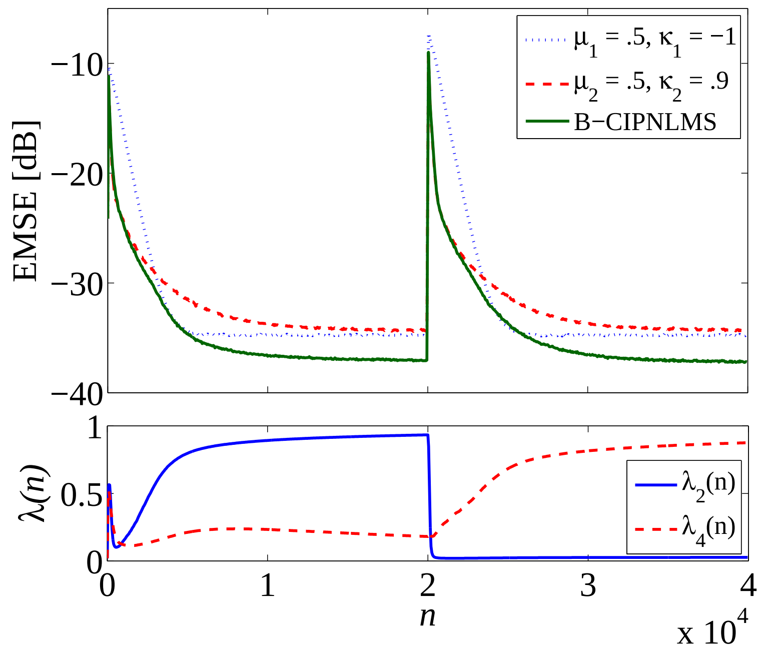


Fig. 9. Performance of the B-CIPNLMS filter ($\mu_1 = \mu_2 = 0.5$, $\kappa_1 = -1$ and $\kappa_2 = 0.9$) when the real echo channel is used. The input signal is WGN. The EMSEs of the IPNLMS components are also included as a reference for comparisons. The lower plot shows the evolution of mixing coefficients $\lambda_2(n)$ and $\lambda_4(n)$.

sometimes we have observed larger gains; for instance, when using the strongly sparse echo channel of Fig. 5, B-CIPNLMS reduced the steady-state EMSE of the components about 7 dB. As for the speed of convergence, we have checked that the selection of M_b has an almost neglegible effect on the convergence of the combined filter.

We have compared the performance of B-CIPNLMS to those of SC-IPNLMS and the improved IPNLMS (IIPNLMS) filter of [3], [4]. The working principle of the IIPNLMS filter is to explicitly classify the coefficients as active or non-active, using a different κ for the coefficients in each group. IIPNLMS introduces several parameters and thresholds which have been adjusted as in [3]. To get a fair comparison with B-CIPNLMS, the IIPNLMS filter adapts the active and non-active coefficients with $\kappa = -1$ and $\kappa = 0.9$, respectively. As for the SC-IPNLMS filter, we have used again $\mu_{\rm SC} = 0.5$ and $\alpha = -0.5$.

Fig. 11 displays EMSE evolution for the three algorithms under consideration when using the real echo channel, and USASI noise for the far-end signal. We can see that B-CIPNLMS and SC-IPNLMS have similar convergence rates, clearly outperforming IIPNLMS both initially, and after the change in the echo path. With respect to steady-state performance, B-CIPNLMS gets a smaller misadjustment than any of the other two echo cancelers. Further insight about the working principles of B-CIPNLMS and its

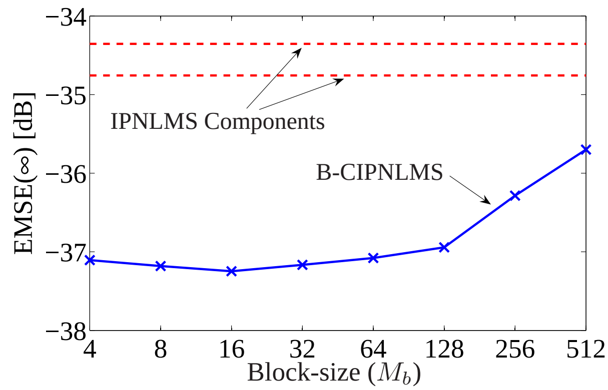


Fig. 10. Steady-state EMSE of the B-CIPNLMS filter ($\mu_1 = \mu_2 = 0.5$, $\kappa_1 = -1$ and $\kappa_2 = 0.9$) as a function of the block-size (M_b).

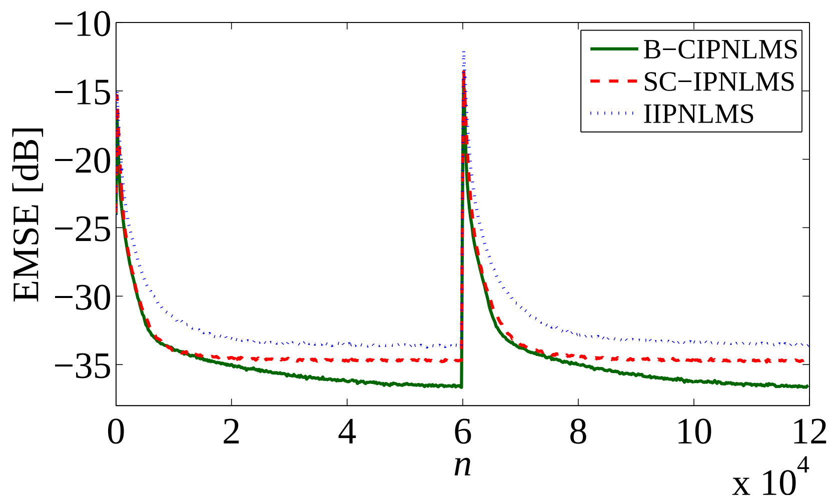


Fig. 11. EMSE performance of B-CIPNLMS, SC-IPNLMS (algorithm from [29]), and IIPNLMS (algorithm from [3], [4]), when the real echo channel is used and the far-end signal is USASI noise.

superior steady-state behavior will be provided by the theoretical analysis carried out in the Appendix.

E. Experiments with speech as input

To conclude the experimental evaluation of the proposed schemes for echo cancellation, in this subsection we will illustrate their performance when the input signal is a fragment of 7 s of real speech, sampled at 8 kHz. Since we are primarily interested on the robustness of the combined filters to channels with different degrees of sparseness, we will again use the dispersive and strongly sparse echo paths depicted in Fig. 5, switching from the former to the latter at $t = 3.5$ s. As before, the power of the output noise has been selected to get an average SNR of 20 dB in the reference signal.

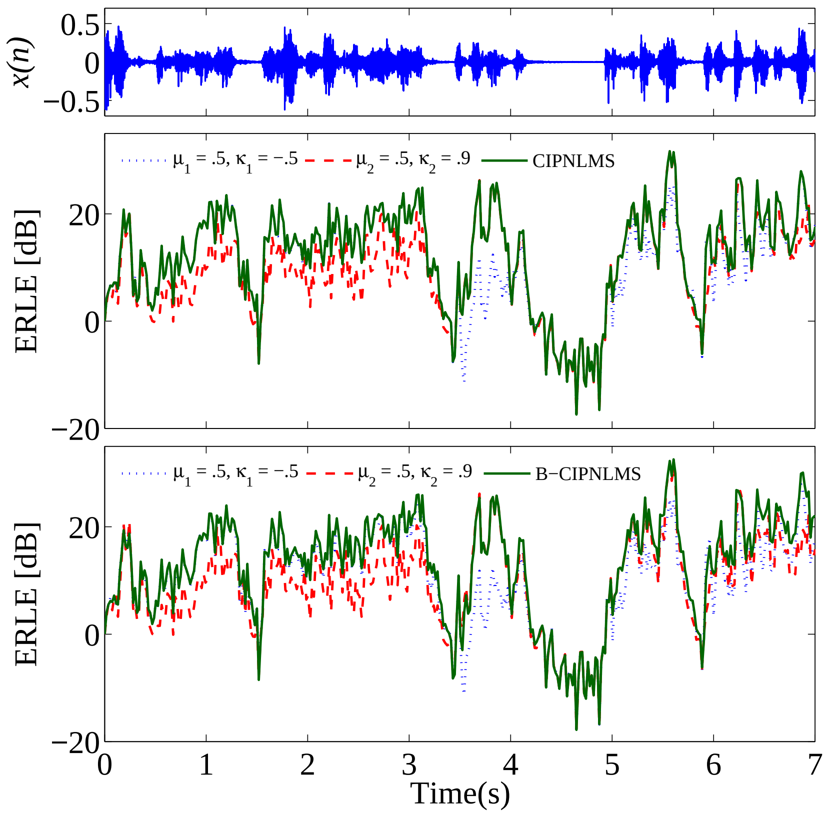


Fig. 12. CIPNLMS and B-CIPNLMS cancellation performance when the far-end signal is a fragment of real speech (represented in the upper panel). At $t = 3.5$ s the echo channel changes and the number of active coefficients is reduced from 256 to only 16. The ERLE of the component IPNLMS filters (parameter settings shown in the legend) has also been represented as a reference for comparison.

In this case, the Echo Return Loss Enhancement (ERLE) is used as the figure of merit:

$$\mathrm{ERLE}(n) = \frac{E\{[d(n) - e_0(n)]^2\}}{E\{[e(n) - e_0(n)]^2\}},$$

estimating both expectations from averages over 300 runs of the algorithms.

In Fig. 12 we represent the evolution of the ERLE of both combined algorithms, together with those of their constituent filters (parameter settings are given in the legend of the figure). We can describe CIPNLMS and B-CIPNLMS performance in very similar terms to those used for WGN and USASI input signals. Depending on the degree of sparsity of the channel, the first or the second IPNLMS components achieve higher ERLEs, and both combination schemes behave, at each iteration, at least as the best performing algorithm. These results reinforce our previous conclusions about the suitability of CIPNLMS and B-CIPNLMS to build echo cancelers with fast convergence and improved robustness to channel sparsity.

V. Conclusions

Proportionate schemes offer better behavior than standard adaptive filters for the cancellation of sparse echo-path impulse responses, but the selection of their parameters is subject to different compromises. In this paper we have shown how combination schemes can help to improve the performance of proportionate filters, by alleviating the speed vs precision tradeoff, as well as by increasing robustness to channels with different degrees of sparsity. A new block-based combination has also been introduced to further reduce the steady-state misadjustment. The performance of such combination schemes has been illustrated when combining IPNLMS filters with different settings, showing in all cases significant advantages over the use of a single filter. Furthermore, a steady-state analysis is carried out in the Appendix that follows, showing that the combination can achieve *better-than-universal* performance, i.e., it can simultaneously outperform both component filters.

Appendix I

Mean-square performance analysis of the proposed algorithms

We present here some theoretical results about the proposed echo cancellation schemes; in particular we give a theoretical justification of the improved steady-state properties of both the CIPNLMS and B-CIPNLMS algorithms, and of the fact that they can outperform both component filters simultaneously.

A. Stationary data model and definitions

To carry out the analysis, we will rely on several simplifying assumptions:

A1. $d(n)$ and $\mathbf{x}(n)$ satisfy a linear regression model $d(n) = \mathbf{w}_{\rm o}^T\mathbf{x}(n) + e_0(n)$, where $\mathbf{w}_{\rm o}$ is an unknown vector of length M, and $e_0(n)$ is a zero mean i.i.d. Gaussian noise with variance σ_0^2, independent of $\mathbf{x}(m), \forall m, n$.

A2. The input signal is a white noise with zero mean and variance σ_x^2. Therefore, the autocorrelation matrix of the input regressors is $\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^T(n)\} = \sigma_x^2\mathbf{I}$.

A3. The step sizes of both IPNLMS filters are small enough so that we can accept that the weight vectors follow the average statistics of the input signal.

Even though several of these assumptions might not be well-justified in practice, they have been extensively used in the adaptive filtering literature, including the theoretical study of proportionate schemes (see, among others, [8], [9], [10], [12]), since they give good insight about the working principles of adaptive filters and normally lead to results that match reasonably well their actual performance. Note that

Assumption A2 does not imply independence among regressors $\{\mathbf{x}(n)\}$, which, in spite of being a very common assumption for filter analysis, is very unrealistic for tapped-delay line filters in which adjacent regressors have several common entries. Instead, the small step-size assumption A3, originally adopted in [5], will be used here.

The following error quantities will appear in the analysis:

- Weight errors: $\boldsymbol{\varepsilon}_i(n) = \mathbf{w}_o - \mathbf{w}_i(n),\ i = 1, 2$, for the component filters, and $\boldsymbol{\varepsilon}(n) = \mathbf{w}_o - \mathbf{w}(n)$ for their combination.
- Filter errors: $e_i(n) = d(n) - y_i(n)$ for the component filters, and $e(n) = d(n) - y(n)$ for the combination.
- *A priori* errors: $e_{a,i}(n) = \boldsymbol{\varepsilon}_i^T(n)\mathbf{x}(n)$ and $e_a(n) = \boldsymbol{\varepsilon}^T(n)\mathbf{x}(n)$. It is easy to check that filter and *a priori* errors are related via $e_{a,i}(n) = e_i(n) - e_0(n)$ and $e_a(n) = e(n) - e_0(n)$.
- Excess mean square errors (EMSE): $J_{\text{ex},i}(n) = E\{[e_i(n) - e_0(n)]^2\} = E\{e_{a,i}^2(n)\}$ and $J_{\text{ex}}(n) = E\{[e(n) - e_0(n)]^2\} = E\{e_a^2(n)\}$.
- Cross-EMSE [18]: $J_{\text{ex},12}(n) = E\{e_{a,1}(n)e_{a,2}(n)\}$.

B. CIPNLMS steady-state performance

1) Universality of the adaptive combination scheme: In [18] we studied the mean-square performance of the convex combination of two adaptive filters, showing that it enjoys *universal capabilities* in steady-state, i.e., that it behaves as the best component filter or, under certain conditions, better than any of them. To be more concrete, we found that the scheme could operate in one of the three following states:

Case 1) $J_{\text{ex},1}(\infty) \leq J_{\text{ex},12}(\infty) \leq J_{\text{ex},2}(\infty)$, where $J_{\text{ex},i}(\infty)$ and $J_{\text{ex}}(\infty)$ are the steady-state values of the EMSE, i.e., their limiting values as $n \to \infty$. In this case, the combination satisfies $J_{\text{ex}}(\infty) = J_{\text{ex},1}(\infty)$.

Case 2) $J_{\text{ex},2}(\infty) \leq J_{\text{ex},12}(\infty) \leq J_{\text{ex},1}(\infty)$. Again, the combination performs like the best component filter, with $J_{\text{ex}}(\infty) = J_{\text{ex},2}(\infty)$.

Case 3) $J_{\text{ex},12}(\infty) < \min\{J_{\text{ex},1}(\infty), J_{\text{ex},2}(\infty)\}$. In this case, the overall EMSE can be approximated as (cf. [18, Eq. (33)])

$$J_{\text{ex}}(\infty) \approx J_{\text{ex},12}(\infty) + \frac{[J_{\text{ex},1}(\infty) - J_{\text{ex},12}(\infty)][J_{\text{ex},2}(\infty) - J_{\text{ex},12}(\infty)]}{J_{\text{ex},1}(\infty) + J_{\text{ex},2}(\infty) - 2J_{\text{ex},12}(\infty)}, \tag{14}$$

and it can be shown that $J_{\text{ex}}(\infty) < \min\{J_{\text{ex},1}(\infty), J_{\text{ex},2}(\infty)\}$. Since in this situation the combination outperforms both components, we say that it presents a *better-than-universal* behavior.

This case appears when the cross-correlation between the *a priori* errors of the component filters is sufficiently small, making possible a reduction of the overall filter error variance.

It is important to remark that the above result holds independently of the component filters. Among the two possibilities for combining IPNLMS filters that were discussed in Section II, we will focus here in the configuration for improving IPNLMS robustness to different degrees of sparsity, for which we will find that the *better-than-universal* behavior is encountered. Therefore, in the following we assume $\mu_1 = \mu_2 = \mu$ and $\kappa_1 < \kappa_2$.

2) Steady-state EMSE of the component filters: IPNLMS filters are adapted according to Eqs. (3)-(5), that we reproduce here for convenience:

$$w_{m,i}(n+1) = w_{m,i}(n) + \mu_{m,i}(n) e_i(n) x_m(n), \tag{15}$$

$$\mu_{m,i}(n) = \frac{\mu\, g_{m,i}(n)}{\delta + \sum_{k=1}^{M} g_{k,i}(n) x_k^2(n)}, \tag{16}$$

$$g_{m,i}(n) = (1-\kappa_i)\frac{1}{2M} + (1+\kappa_i)\frac{|w_{m,i}(n)|}{\epsilon + 2\|\mathbf{w}_i(n)\|_1}, \tag{17}$$

for $m = 1, \ldots, M$ and $i = 1, 2$.

We start by replacing $\sum_{k=1}^{M} g_{k,i}(n) x_k^2(n)$ in (16) by its expected value, $\sigma_x^2 \sum_{k=1}^{M} g_{k,i}(n)$, which is a resonable approximation whenever the variance of the original sum is small. It can be shown (see [30, Sec. 3]) that for white and Gaussian input signals this condition is satisfied if $\sum_{k=1}^{M} g_{k,i}(n) \gg \sqrt{2\sum_{k=1}^{M} g_{k,i}^2(n)}$, which is normally true for impulse responses $\mathbf{w}_{\rm o}$ with several active coefficients[4]. Since, for small ϵ, we have $\sum_{k=1}^{M} g_{k,i}(n) \approx 1$, we can rewrite the update equation of each of the filter weights in the form

$$w_{m,i}(n+1) = w_{m,i}(n) + \frac{\mu}{\sigma_x^2} g_{m,i}(n) e_i(n) x_m(n), \tag{18}$$

where we have assumed a vanishingly small δ.

Next, we subtract the above expression from the mth component of the optimal weight vector, $w_{m,{\rm o}}$. By using also $e_i(n) = \boldsymbol{\varepsilon}_i^T(n)\mathbf{x}(n) + e_0(n)$, and after some manipulations, we arrive at

$$\varepsilon_{m,i}(n+1) = \varepsilon_{m,i}(n) - \frac{\mu}{\sigma_x^2} g_{m,i}(n) \boldsymbol{\varepsilon}_i^T(n)\mathbf{x}(n) x_m(n) - \frac{\mu}{\sigma_x^2} g_{m,i}(n) e_0(n) x_m(n), \tag{19}$$

where $\varepsilon_{m,i}(n)$ denotes the mth component of $\boldsymbol{\varepsilon}_i(n)$. Assumption A3 and the Direct Averaging Method [5] can be applied to replace $\mathbf{x}(n) x_m(n)$ by its expected value in the above expression. For white input

[4] Note that gain factors are between 0 and 1, and they sum to approximately one.

signals, $E\{x_k(n)x_m(n)\} = \sigma_x^2\delta_{km}$, where δ_{km} is the Kronecker delta function, so that (19) simplifies to

$$\varepsilon_{m,i}(n+1) = [1-\mu g_{m,i}(n)]\varepsilon_{m,i}(n) - \frac{\mu}{\sigma_x^2} g_{m,i}(n)e_0(n)x_m(n). \quad (20)$$

Evaluation of the EMSE from the above expression is, in general, very difficult, since $g_{m,i}(n)$ is itself a function of the filter weights. If we are only interested in steady-state performance, some useful results can be derived assuming the final gain factors as constant and equal to those corresponding to the optimal solution,

$$\bar{g}_{m,i} = (1-\kappa_i)\frac{1}{2M} + (1+\kappa_i)\frac{|w_{m,\mathrm{o}}|}{\epsilon + 2\|\mathbf{w}_\mathrm{o}\|_1}. \quad (21)$$

Squaring (20) and taking expectations of the resulting expression, we get

$$E\{\varepsilon_{m,i}^2(n+1)\} = [1-\mu\bar{g}_{m,i}]^2 E\{\varepsilon_{m,i}^2(n)\} + \frac{\mu^2\sigma_0^2}{\sigma_x^2}\bar{g}_{m,i}^2, \quad (22)$$

where we have also used the fact that, in virtue of Assumption A1, $e_0(n)$ is independent of $\boldsymbol{\varepsilon}_i(n)$.

Finally, we can take the limiting value of the above expression as $n \to \infty$. Since steady-state operation is characterized by $\lim_{n\to\infty} E\{\varepsilon_{m,i}^2(n+1)\} = \lim_{n\to\infty} E\{\varepsilon_{m,i}^2(n)\}$, the expected square error of the mth filter tap is given by

$$E\{\varepsilon_{m,i}^2(\infty)\} = \lim_{n\to\infty} E\{\varepsilon_{m,i}^2(n)\} = \frac{\mu\sigma_0^2}{\sigma_x^2}\Psi_{m,i}(\infty), \quad (23)$$

where we have defined

$$\Psi_{m,i}(\infty) = \frac{\bar{g}_{m,i}}{2-\mu\bar{g}_{m,i}}. \quad (24)$$

In order to derive an expression for the EMSE of the components, we need to rely on the approximation

$$J_{\mathrm{ex},i}(n) = E\{e_{a,i}^2(n)\} \approx E\{\boldsymbol{\varepsilon}_i^T(n)\mathbf{R}\boldsymbol{\varepsilon}_i(n)\} = \sigma_x^2 \sum_{m=1}^{M} E\{\varepsilon_{m,i}^2(n)\}, \quad (25)$$

which is obtained by applying A3 and the Direct Averaging Method, as explained in [5]. Thus, the steady-state EMSE of the IPNLMS filters can be estimated as

$$J_{\mathrm{ex},i}(\infty) = \mu\sigma_0^2 \sum_{m=1}^{M} \Psi_{m,i}(\infty). \quad (26)$$

3) Steady-state cross-EMSE: We start by multiplying (20) by itself, for $i=1$ and $i=2$. Replacing the gain factors by their optimal values, and taking the expectation of the result, we arrive at

$$E\{\varepsilon_{m,1}(n+1)\varepsilon_{m,2}(n+1)\} = [1-\mu\bar{g}_{m,1}][1-\mu\bar{g}_{m,2}]E\{\varepsilon_{m,1}(n)\varepsilon_{m,2}(n)\} + \frac{\mu^2\sigma_0^2}{\sigma_x^2}\bar{g}_{m,1}\bar{g}_{m,2}. \quad (27)$$

Proceeding as for the IPNLMS filters, taking the limit of this expression as $n \to \infty$ leads to

$$E\{\varepsilon_{m,1}(\infty)\varepsilon_{m,2}(\infty)\} = \lim_{n\to\infty} E\{\varepsilon_{m,1}(n)\varepsilon_{m,2}(n)\} = \frac{\mu\sigma_0^2}{\sigma_x^2}\Psi_{m,12}(\infty), \tag{28}$$

with

$$\Psi_{m,12}(\infty) = \frac{\bar{g}_{m,1}\bar{g}_{m,2}}{\bar{g}_{m,1} + \bar{g}_{m,2} - \mu\bar{g}_{m,1}\bar{g}_{m,2}}. \tag{29}$$

Taking into consideration that the cross-EMSE of the two filters can be approximated by

$$J_{\text{ex},12}(n) = E\{e_{a,1}(n)e_{a,2}(n)\} \approx E\{\boldsymbol{\varepsilon}_i^T(n)\mathbf{R}\boldsymbol{\varepsilon}_2(n)\} = \sigma_x^2 \sum_{m=1}^{M} E\{\varepsilon_{m,1}(n)\varepsilon_{m,2}(n)\}, \tag{30}$$

the steady-state cross-EMSE of the two IPNLMS filters can be expressed as

$$J_{\text{ex},12}(\infty) = \mu\sigma_0^2 \sum_{m=1}^{M} \Psi_{m,12}(\infty). \tag{31}$$

4) Steady-state EMSE of the CIPNLMS filter: Using the expressions we have just derived for the steady-state EMSE and cross-EMSE of the component filters [equations (26) and (31), respectively], the steady-state performance of the CIPNLMS scheme can be modelled using the results for the three cases described in Part 1 of Subsection I-B.

We have studied the performance of a convex combination of IPNLMS filters with fixed parameters $\mu_1 = \mu_2 = \mu = 0.1$ and $\kappa_1 = -1$, as a function of κ_2. The rest of settings of the echo cancellation scenario are as in the experiments section, using both the real echo channel and the artificial strongly sparse channel. Fig. 13 plots (26), for $i = 2$, and (31). The real performance of the filters is also included in the figure, by averaging $e_{a,2}^2(n)$ and $e_{a,1}(n)e_{a,2}(n)$ for 20000 iterations after the convergence of both IPNLMS filters, and over 50 independent runs. The good match between theoretical and simulation results indicate that (26) and (31) provide reasonably good models for the EMSE and cross-EMSE of IPNLMS filters. In particular, it is important to notice that the analysis predicts that $J_{\text{ex},12}(\infty) < J_{\text{ex},i}(\infty),\ i = 1, 2$ (note that the EMSE of the first IPNLMS component can also be read from the figure, for $\kappa_2 = -1$); thus, according to the third case described in Part 1 of this subsection, CIPNLMS should get *better-than-universal* performance.

For the case in which $J_{\text{ex},12}(\infty) < J_{\text{ex},i}(\infty),\ i = 1, 2$, we can use (14) to estimate the steady-state EMSE of the combination; the corresponding results have also been illustrated in Fig. 13. As predicted by the analysis, CIPNLMS outperforms simultaneously both IPNLMS components for a wide range of κ_2, a conclusion that is confirmed by simulation results and explains the *better-than-universal* behavior previously observed in Figs. 6 and 8.

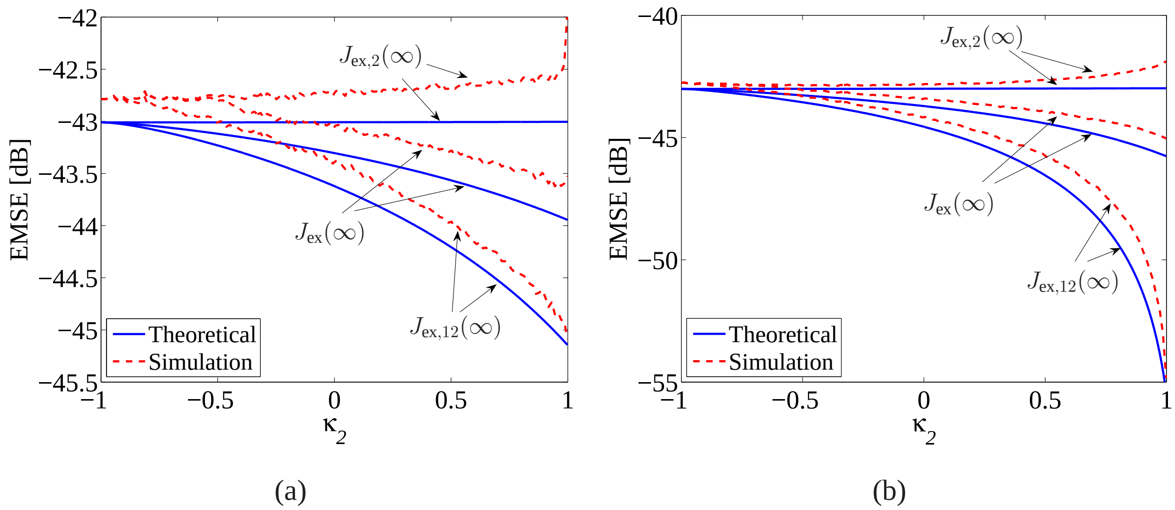


Fig. 13. Steady-state theoretical and estimated EMSE of a CIPNLMS filter when combining two IPNLMS filters with $\mu = 0.1$ and different κ parameters: For the first filter, $\kappa_1 = -1$ is used in all cases, depicting the result as a function of the value of the parameter of the second filter, κ_2. The cross-EMSE between the component filters, and the EMSE of the second component have also been illustrated. (a) The echo path is the real channel depicted in Fig. 3. (b) The echo path is the artificial strongly sparse channel of Fig. 5(b).

For larger values of μ, the deviation between theoretical predictions and simulation results increases (about 1 dB for $\mu = 0.5$), as a consequence of Assumption A3 being less realistic. However, even in this case, the analysis predicts that, as it occurs in the practice, CIPNLMS simultaneously outperforms both its component filters. Assuming Gaussianity of the input regressors, it is possible to derive more accurate expressions for the EMSE and cross-EMSE. However, we have preferred to omit such an assumption in our analysis, so that our results can be applied to more general situations.

C. B-CIPNLMS steady-state performance

As explained in Section III, and suggested also by the analysis of IPNLMS carried out in the previous subsection [see Eq. (26)], in steady-state the gradient noise introduced by each tap of the IPNLMS filter is more significant for active coefficients. Now, we will give theoretical evidence on how the block-based combination scheme can exploit this fact to simultaneously improve the steady state properties of both its component filters.

We start from (25), which allows us to express the overall steady-state EMSE as a sum of terms associated to the different filter weights. Noting also that the B-CIPNLMS scheme can be seen as a filter whose weights are obtained as convex combinations of the weights of the component filters, it follows that

$$\begin{aligned} J_{\rm ex}(n) &= \sigma_x^2 \sum_{m=1}^{M} E\{\varepsilon_m^2(n)\} \\ &= \sigma_x^2 \sum_{l=1}^{L} \sum_{m\in\mathcal{I}_l} E\left\{[\lambda_l(n)\varepsilon_{m,1}(n) + [1-\lambda_l(n)]\varepsilon_{m,2}(n)]^2\right\}. \end{aligned} \tag{32}$$

For $n \to \infty$, we can accept (see [18]) that the steady-state values of the mixing parameters are independent of the weight errors, and the above expression can be rewritten as

$$\begin{aligned} J_{\rm ex}(\infty) &= \sigma_x^2 \sum_{l=1}^{L} \sum_{m\in\mathcal{I}_l} \bar{\lambda}_l^2(\infty)E\{\varepsilon_{m,1}^2(\infty)\} + [1-\bar{\lambda}_l(\infty)]^2 E\{\varepsilon_{m,2}^2(\infty)\} \\ &\qquad + 2\bar{\lambda}_l(\infty)[1-\bar{\lambda}_l(\infty)]E\{\varepsilon_{m,1}(\infty)\varepsilon_{m,2}(\infty)\} \\ &= \mu\sigma_0^2 \sum_{l=1}^{L} \sum_{m\in\mathcal{I}_l} \bar{\lambda}_l^2(\infty)\Psi_{m,1}(\infty) + [1-\bar{\lambda}_l(\infty)]^2\Psi_{m,2}(\infty) + 2\bar{\lambda}_l(\infty)[1-\bar{\lambda}_l(\infty)]\Psi_{m,12}(\infty), \end{aligned} \tag{33}$$

where we have defined $\bar{\lambda}_l(\infty) = \lim_{n\to\infty} E\{\lambda_l(n)\}$. It is clear that, for $\bar{\lambda}_l(\infty) = 1,\ \forall l$, or $\bar{\lambda}_l(\infty) = 0$, $\forall l$, (25) is recovered, and the B-CIPNLMS filter behaves as the first or the second component filter, respectively. By allowing different values for the mixing parameters, and by updating them with the objective to minimize the overall square error [according to (13)], we can keep the estimation of each block of coefficients provided by the component filter which is incurring in a smaller quadratic error for that particular group of weights. Therefore, B-CIPNLMS is expected to incur in a smaller steady-state misalignment than any of the component filters.

This behavior has been illustrated in Figs. 9 and 10 for the block-based combination of an NLMS and a PNLMS filter. From the evolution of mixing parameters $\lambda_2(n)$ and $\lambda_4(n)$, it is evident that the overall filter is combining the steady-state estimation of the echo-path coefficients provided by both components filters and, as a result of this, it achieves a decrement of their steady-state EMSEs.

ACKNOWLEDGEMENTS

The authors thank Dr. J. Benesty for kindly supplying the real echo-channel data.

REFERENCES

[1] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L Gay, *Advances in Network and Acoustic Echo Cancellation*, Berlin: Springer-Verlag, 2001.

[2] C. Breining et. al., "Acoustic Echo Control. An application of very-high-order adaptive filters," in *IEEE Signal Process. Mag.*, vol. 16, pp. 42–69, 1999.

[3] J. Cui, P. A. Naylor, and D. T. Brown, "An improved IPNLMS algorithm for echo cancellation in packet-switched networks," in *Proc. ICASSP'04*, Montreal, Canada, vol. IV, 2004, pp. 141–144.

[4] P. A. Naylor, J. Cui, and M. Brookes, "Adaptive algorithms for sparse echo cancellation," *Signal Processing*, vol. 86, pp. 1182–1192, 2004.

[5] S. Haykin, *Adaptive Filter Theory*, Upper Saddle River, NJ: Prentice-Hall, 2002.

[6] A. H. Sayed, *Fundamentals of Adaptive Filtering*, New York, NY: Wiley, 2003.

[7] S. Makino, Y. Kaneda, and N. Koizumi, "Exponentially weighted step-size NLMS adaptive filter based on the statistics of a room impulse response," *IEEE Trans. Speech Audio Process.*, vol. 1, pp. 101–108, 1993.

[8] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Signal Process.*, vol. 8, pp. 508–518, 2000.

[9] H. Deng and M. Doroslovacki "Improving convergence of the PNLMS algorithm for sparse impulse response identification," *IEEE Signal Process. Lett.*, vol. 12, pp. 181–184, 2005.

[10] H. Deng and M. Doroslovacki "Proportionate adaptive algorithms for network echo cancellation," *IEEE Trans. Signal Process.*, vol. 54, pp. 1794–1803, 2006.

[11] S. L. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, vol. 1, 1998, pp. 394–398.

[12] M. Nekuii and M. Atarodi, "A fast converging algorithm for network echo cancellation," *IEEE Signal Process. Lett.*, vol. 11, pp. 427–430, 2004.

[13] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. ICASSP'02*, Orlando, FL, vol. II, 2002, pp. 1881–1883.

[14] J. Kivinen and M. K. Warmuth "Exponentiated gradient versus gradient descent for linear predictors," *Inform. Comput.*, vol. 132, pp. 1–64, 1997.

[15] J. Benesty and Y. Huang, "The LMS, PNLMS, and exponentiated gradient algorithms," in *Proc. European Signal Process. Conf.*, Vienna, Austria, 2004, pp. 721–724.

[16] J. Arenas-García, M. Martínez-Ramón, A. Navia-Vázquez, and A. R. Figueiras-Vidal, "Plant identification via adaptive combination of transversal filters," *Signal Processing*, vol. 86, pp. 2430–2438, 2006.

[17] J. Arenas-García, V. Gómez-Verdejo, and A. R. Figueiras-Vidal, "New algorithms for improved adaptive convex combination of LMS transversal filers," *IEEE. Trans. Instr. Meas.*, vol. 54, pp. 2239–2249, 2005.

[18] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, 1078–1090, 2006.

[19] M. T. M. Silva and V. H. Nascimento, "Improving the tracking capabilities of adaptive filters via convex combination," *IEEE Trans. Signal Process.*, vol. 56, 3137–3149, 2008.

[20] D. P. Mandic, P. Vayanos, S. Javidi, B. J. Jelfs, and K. Aihara, "Online tracking of the degree of nonlinearity within complex signals," in *Proc. ICASSP'08*, Las Vegas, NV, 2008, pp. 2061–2064.

[21] J. Arenas-García and A. R. Figueiras-Vidal, "Adaptive combination of IPNLMS filters for robust sparse echo cancellation," in *Proc. IEEE Intl. Workshop on Mach. Learning for Signal Process.*, Cancun, Mexico, 2008.

[22] A. W. H. Khong and P. A. Naylor, “Selective-tap adaptive filtering with performance analysis for the identification of time-varying systems,” *IEEE Trans. Audio, Speech and Language Process.*, vol. 15, 1681–1695, 2007.

[23] A. W. H. Khong and P. A. Naylor, “Stereophonic acoustic echo cancellation employing selective-tap adaptive algorithms,” *IEEE Trans. Audio, Speech and Language Process.*, vol. 14, 785–796, 2007.

[24] T. Aboulnasr and K. Mayyas, “Complexity reduction of the NLMS algorithm via selective partial updates,” *IEEE Trans. Signal Process.*, vol. 47, 1421–1424, 1999.

[25] Standard ANSI/EIA-549-88, “NRSC AM Preemphasis/Deemphasis and Broadcast Audio Transmission Bandwidth Specifications (ANSI/EIA-549-88),” 1988.

[26] T. Gänsler, S. L. Gay, M. Sondhi, and J. Benesty, “Double-talk robust fast converging algorithms for network echo cancellation,” *IEEE Trans. Speech Audio Process.*, vol. 8, pp. 656–663, 2000.

[27] A. Mader, H. Puder, and G. U. Schmidt, “Step-size control for acoustic echo cancellation filter – an overview,” *Signal Processing*, vol. 80, pp. 1697–1719, 2000.

[28] J.-M. Valin, “On adjusting the learning rate in frequency domain echo cancellation with double talk,” *IEEE Trans. Audio, Speech and Language Process.*, vol. 15, 1030–1034, 2007.

[29] A. W. H. Kong and P. A. Naylor, “Efficient use of sparse adaptive filters,” in *Proc. 40th Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, 2006, pp. 1375–1379.

[30] M. Doroslovacki and H. Deng, “On convergence of proportionate-type NLMS adaptive algorithms,” in *Proc. ICASSP’06*, vol. III, Toulouse, France, 2006, pp. 105–108.