Combinations of Adaptive Filters

Jerónimo Arenas-García, *Senior Member, IEEE*, Luis A. Azpicueta-Ruiz, *Member, IEEE*,

Magno T. M. Silva, *Member, IEEE*, Vítor H. Nascimento, *Senior Member, IEEE*,

Ali H. Sayed, *Fellow, IEEE*

Adaptive filters are at the core of many signal processing applications, ranging from acoustic noise supression to echo cancellation [1], array beamforming [2], channel equalization [3], and to more recent sensor network applications in surveillance, target localization and tracking. A trending approach in this direction is to recur to in-network distributed processing in which individual nodes implement adaptation rules and diffuse their estimation to the network [4], [5].

Ranging from the simple Least-Mean-Squares (LMS) to sophisticated state-space algorithms, significant research has been carried out over the last 50 years to develop effective adaptive algorithms, in an attempt to improve their general properties in terms of convergence, tracking ability, steady-state misadjustment, robustness, or computational cost [6]. Many design procedures and theoretical models have been developed, and many novel adaptive structures are continually proposed with the objective of improving filter behavior with respect to well-known performance tradeoffs (such as convergence rate *versus* steady-state performance), or incorporating available *a priori* knowledge into the learning mechanisms of the filters (e.g., to enforce sparsity).

When the *a priori* knowledge about the filtering scenario is limited or imprecise, selecting the most adequate filter structure and adjusting its parameters becomes a challenging task, and erroneous choices can lead to inadequate performance. To address this difficulty, one useful approach is to rely on combinations of adaptive structures. Combinations of adaptive schemes have been studied in several works [7]–[21] and have been applied to a variety of applications such as the characterization of signal modality [22], acoustic echo cancellation [23]–[26], adaptive line enhancement [27], array beamforming [28], [29], and active noise control [30], [31].

The combination of adaptive filters exploits to some extent the same *divide and conquer* principle that has also been successfully exploited by the machine learning community (e.g., in bagging or boosting [32]). In particular, the problem of combining the outputs of several learning algorithms (*mixture-of-experts*) has been studied in the computational learning field under a different perspective: rather than studying the expected performance of the mixture, deterministic bounds are derived that apply to individual sequences

and, therefore, reflect worst-case scenarios [33]–[35]. These bounds require assumptions different from the ones typically used in adaptive filtering, which is the emphasis of this overview article. In this work, we review the key ideas and principles behind these combination schemes, with emphasis on design rules. We also illustrate their performance with a variety of examples.

I. PROBLEM FORMULATION AND NOTATION

We generally assume the availability of a reference signal and an input regressor vector, $d(n)$ and $\mathbf{u}(n)$, respectively, which satisfy a linear regression model of the form:

$$d(n) = \mathbf{u}^\top(n)\mathbf{w}_\mathrm{o}(n) + v(n), \tag{1}$$

where $\mathbf{w}_\mathrm{o}(n)$ represents the (possibly) time varying optimal solution, and $v(n)$ is a noise sequence, which is considered i.i.d., and independent of $\mathbf{u}(m)$, for all n, m. In this paper, we will mostly restrict ourselves to the case in which all involved variables are real, although the extension to the complex case is straightforward, and has been analyzed in other works [6]. In order to estimate the optimal solution at time n, adaptive filters typically implement a recursion of the form

$$\mathbf{w}(n + 1) = \mathbf{f}\left[\mathbf{w}(n), d(n), \mathbf{u}(n), \mathbf{s}(n)\right],$$

where different adaptive schemes are characterized by their update functions $\mathbf{f}[\cdot]$, and $\mathbf{s}(n)$ represents any other state information that is needed for the update of the filter.

We define the following error variables that are commonly used to characterize the performance of adaptive filters [6]:

- Filter output: $y(n) = \mathbf{u}^\top(n)\mathbf{w}(n)$
- Weight error: $\widetilde{\mathbf{w}}(n) = \mathbf{w}_\mathrm{o}(n) - \mathbf{w}(n)$
- *A priori* filter error: $e_a(n) = \mathbf{u}^\top(n)\widetilde{\mathbf{w}}(n)$
- Filter error: $e(n) = d(n) - \mathbf{u}^\top(n)\mathbf{w}(n) = e_a(n) + v(n)$
- Mean Square Error: $\mathrm{MSE}(n) = \mathbb{E}\{e^2(n)\}$
- Excess MSE (EMSE): $\zeta(n) = \mathbb{E}\{e_a^2(n)\} = \mathrm{MSE}(n) - \mathbb{E}\{v^2(n)\}$
- Mean Square Deviation: $\mathrm{MSD}(n) = \mathbb{E}\{\|\widetilde{\mathbf{w}}(n)\|^2\}$

During their operation, adaptive filters normally go from a convergence phase, where the expected mean-square error decreases, to a steady-state regime in which the mean-square error tends towards some asymptotic value. Thus, for steady-state performance we also define the steady-state MSE, EMSE, and MSD as their limiting values as n increases. For instance, the steady-state EMSE is defined as

$$\zeta(\infty) = \lim_{n \to \infty} \mathbb{E}\{e_a^2(n)\}.$$

II. A BASIC COMBINATION OF TWO ADAPTIVE FILTERS

The most simple combination scheme incorporates two adaptive filters. Figure 1 illustrates the configuration, which shows that combination schemes have two concurrent adaptive layers: adaptation of individual filters and adaptation of the combination layer. As illustrated in the figure, both adaptive filters have access to the same input and reference signals and produce their individual estimates of the optimum weight vector, $\mathbf{w}_o(n)$. The goal of the combination layer is to learn which filter component is performing better dynamically at any particular time, assigning them weights to optimize the overall performance.
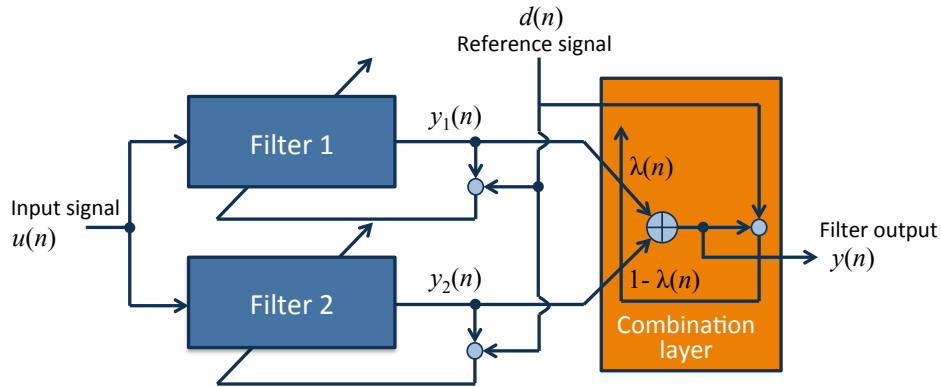

Fig. 1.   Basic scheme for the combination of two adaptive filters.

In this paper, we focus on affine and convex combinations, where the overall filter output is given by

$$y(n) = \lambda(n)y_1(n) + [1 - \lambda(n)]y_2(n), \tag{2}$$

where $y_i(n) = \mathbf{u}^\top(n)\mathbf{w}_i(n)$, $i = 1, 2$, are the outputs ot the two adaptive filters characterized by weights $\mathbf{w}_i(n)$, and $\lambda(n)$ is a mixing parameter. Similarly, the estimated weight vector, the error, and the *a priori* error of the combination scheme are given by

$$\mathbf{w}(n) = \lambda(n)\mathbf{w}_1(n) + [1 - \lambda(n)]\mathbf{w}_2(n), \tag{3}$$

$$e(n) = \lambda(n)e_1(n) + [1 - \lambda(n)]e_2(n), \tag{4}$$

$$e_a(n) = \lambda(n)e_{a,1}(n) + [1 - \lambda(n)]e_{a,2}(n), \tag{5}$$

where $e_i(n) = d(n) - y_i(n)$, $i = 1, 2$, are the errors of the two adaptive filter components, and $e_{a,i}(n)$ their *a priori* errors. The combination parameter $\lambda(n)$ is constrained to lie in the interval $[0, 1]$ for convex combinations, while it can take any real value for affine combinations. There are also combinations where the sum of weights assigned to the component filters is not equal to one [10]. These unconstrained combinations

will not be addressed in this paper. In order to obtain an enhanced behavior from the combination, it is crucial to design mechanisms to learn the mixing parameter $\lambda(n)$ and to adapt it to track drifts in the model. Different algorithms for the adaptation of the combination have been proposed in the literature; some of them will be reviewed in a later section.

In (3), we are implicitly assuming component filters with the same length. When this is not the case, we can still use (3) by extending the shortest filter and filling with zeros the nonexistent coefficients. Using this observation, most of the algorithms discussed in this paper continue to be valid if the component filters have different lengths. Moreover, in practice, construction (3) does not need to be evaluated since only the output combination (2) is necessary. Furthermore, although not required by the analysis, in the experiments we will assume that the component filter weight vectors match the length of the optimum solution, $\mathbf{w}_\mathrm{o}(n)$. Nevertheless, there have been useful studies in the literature where the length of the filters is treated as a design variable and the adaptation rules are used to learn the length of the optimum solution as well to avoid degradation due to under-modeling or over-modeling [9], [13], [19].

The selection of the individual filters is often influenced by the application itself. For example, combination structures can be used to:

- Facilitate the selection of filter parameters (such as the step size or forgetting factor, regularization constants, filter length, projection order, etc). Existing schemes fix the values of these parameters or recur to complex schemes that learn them over time. Alternatively, we can consider a combination structure where the two filters belong to the same family of algorithms, but differ in the values of the filter parameters. The combination layer would then select and give more weight to the best filter at every time instant, making parameter selection less critical.

- Increase robustness against unknown environmental conditions. A common objective in the design of adaptive filters consists in the incorporation of *a priori* knowledge about the filtering scenario to improve the performance of the filter. For instance, sparse-aware adaptive filters try to benefit from the kwnowledge that many coefficients of the optimal solution are (close to) zero, and Volterra filters try to model non-linearities with polynomial kernels. However, when the filtering conditions are not known accurately, or change over time, these schemes can show very suboptimal performance. Combination schemes can be exploited to increase the robustness of these techniques against lack of or imprecise knowledge of the adaptive filtering scenario.

- Provide diversity that can be exploited to enhance performance beyond the capabilities of an individual adaptive filter. An example will be given later, when we study the tracking abilities of a combination of one LMS and one recursive least-squares (RLS) filters.

The main disadvantage of this approach is the increased computational cost required for the adaptation of

the two adaptive filters. However, it should be noted that the filters can be adapted in parallel and some strategies allow for the joint adaptation of the two filters with just a slight increment with respect to that of a single adaptive filter. Regarding the adaptation of the mixing parameter, the computational requirement of most available schemes is not significant.

In the next section, we review the theoretical limits of the combination scheme consisting of just two filters. Then, we review different combination rules that have already appeared in the literature, and compare them in a variety of simulated conditions.

III. OPTIMUM MIXING PARAMETER AND COMBINATION PERFORMANCE

In this section, we derive the expression for the optimal mixing parameter in the affine and convex cases, in the sense of minimizing the MSE of the combination. Since for $\lambda(n) = 0$ and $\lambda(n) = 1$ convex and affine combinations are equivalent to each of the component filters, we know that an optimal selection of the mixing parameter would guarantee that these combinations perform at least as the best component filter. In this section, we examine the behavior of the combination scheme for other non-trivial choices of the mixing parameter.

To start with, it can be easily seen that the EMSE of the combination (2) is given by

$$\zeta(n) = \mathbb{E}\{e_a^2(n)\} = \lambda^2(n)\zeta_1(n) + [1 - \lambda(n)]^2\zeta_2(n) + 2\lambda(n)[1 - \lambda(n)]\zeta_{12}(n), \tag{6}$$

where $\zeta_i(n)$, $i = 1, 2$, are the EMSEs of the two individual filters, and $\zeta_{12}(n)$ denotes the cross-EMSE defined as [12]:

$$\zeta_{12}(n) = \mathbb{E}\{e_{a,1}(n)e_{a,2}(n)\}.$$

This cross-EMSE variable is an important measure and is closely related to the ability of the combination to improve the performance of both filter components, as we will see shortly. An important property of the cross-EMSE that will be useful when discussing the properties of the optimal combination is derived from the Cauchy-Schwartz inequality, namely:

$$|\zeta_{12}(n)|^2 \le \zeta_1(n)\zeta_2(n), \tag{7}$$

which implies that the cross-EMSE can never be simultaneously larger (in absolute terms) than both individual EMSEs.

Expression (6) reveals a quadratic dependence on the mixing factor, $\lambda(n)$. It follows that, in the affine case, the optimal choice for $\lambda(n)$ is given by:

$$\lambda_{\text{aff}}^{\text{o}}(n) = \frac{\zeta_2(n) - \zeta_{12}(n)}{\zeta_1(n) + \zeta_2(n) - 2\zeta_{12}(n)} = \frac{\Delta\zeta_2(n)}{\Delta\zeta_1(n) + \Delta\zeta_2(n)}, \tag{8}$$

where we have defined $\Delta\zeta_i(n) = \zeta_i(n) - \zeta_{12}(n)$, $i = 1, 2$. In the case of a convex combination, the minimization of (6) needs to be carried out by constraining $\lambda(n)$ to lie in the interval $[0, 1]$. Given that (6) is non-negative and quadratic in $\lambda(n)$, the optimum mixing parameter is either given by (8) or lies in the limits of the considered interval, i.e.,

$$\lambda_{\text{cvx}}^{\text{o}}(n) = \left.\frac{\Delta\zeta_2(n)}{\Delta\zeta_1(n) + \Delta\zeta_2(n)}\right|_0^1, \tag{9}$$

where the vertical line denotes truncation to the indicated values.

Substituting either (8) or (9) into (6), we find that the optimum EMSE of the filter is given by

$$\zeta^{\text{o}}(n) = \zeta_1(n) - [1 - \lambda^{\text{o}}(n)]\,\Delta\zeta_1(n) \tag{10a}$$

$$= \zeta_2(n) - \lambda^{\text{o}}(n)\Delta\zeta_2(n), \tag{10b}$$

where λ^{o} stands for either $\lambda_{\text{aff}}^{\text{o}}(n)$ or $\lambda_{\text{cvx}}^{\text{o}}(n)$. Note that these expressions are valid for any time instant n, and also for the steady-state performance of the filter (i.e., for $n \to \infty$).

The analysis of the expressions for the optimum mixing parameter and for the combination EMSE allows us to conclude that, depending on the cross-EMSE value, the combination can be performing in one of four possible conditions at every time instant. Taking into account that the denominator of (8) is non-negative, $\Delta\zeta_1(n) + \Delta\zeta_2(n) = \mathbb{E}\{[e_{a,1}(n) - e_{a,2}(n)]^2\}$, Table I summarizes the main properties of these four possible cases (the time index is omitted in the first column for the sake of compactness).

TABLE I

POSSIBLE OPERATION REGIMES FOR THE AFFINE AND CONVEX COMBINATION OF TWO ADAPTIVE FILTERS.

	$\zeta_{12}(n)$	$\Delta\zeta_1(n)$	$\Delta\zeta_2(n)$	$\lambda_{\text{aff}}^{\text{o}}(n)$	$\lambda_{\text{cvx}}^{\text{o}}(n)$
Case 1:	$\zeta_1 \leq \zeta_{12}\ (< \sqrt{\zeta_1\zeta_2} < \zeta_2)$	≤ 0	> 0	≥ 1	$= 1$
Case 2:	$\zeta_2 \leq \zeta_{12}\ (< \sqrt{\zeta_1\zeta_2} < \zeta_1)$	> 0	≤ 0	≤ 0	$= 0$
Case 3:	$(-\sqrt{\zeta_1\zeta_2} <)\ \zeta_{12} < \min\{\zeta_1, \zeta_2\}$	> 0	> 0	$\in (0, 1)$	$\in (0, 1)$
Case 4:	$\zeta_1 = \zeta_2 = \zeta_{12}$	$= 0$	$= 0$	—	—

From the information included in the table we can deduce the following properties:

- In cases 1 and 2, the convexity constraints are active. As a consequence, the optimum mixing parameter of the convex combination is either 1 or 0, respectively, making this scheme behave just like the best filter component. However, an affine combination can outperform both components in this case. To see this, consider (10a) and (10b) for cases 1 and 2, respectively, and note that a positive value is subtracted from the smallest filter EMSE (ζ_1 and ζ_2) to obtain the EMSE of the combination.

- In case 3, affine and convex combinations share the same optimum mixing parameter. As in the previous case, using (10a) and (10b) we can easily conclude that, in this case, combinations outperform both component filters simultaneously. This performance can be explained from the small cross-EMSE in this case: since the correlation between the *a priori* errors of both component filters is small, their weighted combination provides an estimation error of reduced variance.

- For completeness, we have included a fourth case where $\zeta_1 = \zeta_2 = \zeta_{12}$. Simplifying all terms that involve $\lambda(n)$ in (6), we can conclude that in this case we have $\zeta^o(n) = \zeta_1(n) = \zeta_2(n)$, irrespectively of the value of the mixing parameter. It is important to remark that for case 4 to hold, condition $\zeta_1 = \zeta_2 = \zeta_{12}$ should be satisfied exactly, so that this case will be rarely encountered in practice.

We should emphasize that the results in this section refer to the theoretical performance of optimally adjusted affine and convex combinations. Practical algorithms to adjust the mixing parameter will be presented in a later section, and will imply the addition of a certain amount of noise that, in many cases, justify the adoption of convexity constraints.

A. Cross-EMSE of LMS and RLS filters

The cross-EMSE plays an important role in clarifying the performance regime of a combination of two adaptive filters. Several works have extended the available theoretical models for the EMSE of adaptive filters to the derivation of expressions for the cross-EMSE [6], [12], [15], [16]. In this section, we review some of the main results for LMS and RLS filters.

Assume that the optimal solution satisfies the following random-walk model[1]

$$\mathbf{w}_o(n) = \mathbf{w}_o(n-1) + \mathbf{q}(n), \tag{11}$$

where $\mathbf{q}(n)$ is the change of the solution at every step that is assumed to have zero-mean and covariance $\mathbf{Q} = \mathbb{E}\{\mathbf{q}(n)\mathbf{q}^\top(n)\}$. It is known [6] that in this case there is a tradeoff between filter performance and tracking ability. The following table shows an approximation of the steady-state EMSE of LMS and RLS adaptive filters derived using the energy conservation approach described in [6]. From these expressions, it is straightforward to obtain the optimum step size for the LMS adaptation rule (μ_o) and the optimal forgetting factor for the RLS adaptive filter with exponentially decaying memory ($1 - \beta_o$). These expressions are also indicated in the table, together with the associated optimum EMSE for each filter family, $\zeta_o(\infty)$.

---

[1]An inconvenience of this model is that it implies divergence of the optimum solution, i.e., $\lim_{n\to\infty} \mathbb{E}\{\|\mathbf{w}_o(n)\|^2\} = \infty$. Nevertheless, this model has been extensively used in the adaptive filtering literature, as it is simpler and leads to similar results than other more realistic tracking models —see [6] for further explanations on this issue.

TABLE II

TRACKING STEADY-STATE PERFORMANCE OF LMS AND RLS ADAPTIVE FILTERS. THE TABLE SHOWS STEADY-STATE EMSE AS A FUNCTION OF THE LMS STEP SIZE (μ) AND RLS FORGETTING FACTOR ($1 - \beta$), THE OPTIMAL VALUES OF THESE PARAMETERS, AND THE MINIMUM EMSE THAT CAN BE ACHIEVED BY BOTH ADAPTIVE FILTERS [$\zeta_o(\infty)$].

Algorithm	$\zeta(\infty)$	μ_o or β_o	$\zeta_o(\infty)$
LMS	$\frac{1}{2}\left[\mu\sigma_v^2\text{Tr}\{\mathbf{R}\} + \mu^{-1}\text{Tr}\{\mathbf{Q}\}\right]$	$\sqrt{\frac{\text{Tr}\{\mathbf{Q}\}}{\sigma_v^2\text{Tr}\{\mathbf{R}\}}}$	$\sqrt{\sigma_v^2\text{Tr}\{\mathbf{R}\}\text{Tr}\{\mathbf{Q}\}}$
RLS	$\frac{1}{2}\left[\beta\sigma_v^2 M + \beta^{-1}\text{Tr}\{\mathbf{QR}\}\right]$	$\sqrt{\frac{\text{Tr}\{\mathbf{QR}\}}{\sigma_v^2 M}}$	$\sqrt{\sigma_v^2 M\text{Tr}\{\mathbf{QR}\}}$

$\sigma_v^2 = \mathbb{E}\{v^2(n)\}$; $\mathbf{Q} = \mathbb{E}\{\mathbf{q}(n)\mathbf{q}^\top(n)\}$; $\mathbf{R} = \mathbb{E}\{\mathbf{u}(n)\mathbf{u}^\top(n)\}$; M is the filter length

In order to study the behavior of combinations of this kind of filters, references [14], [15] derived expressions for the steady-state cross-EMSE, for cases in which the two component filters belong to the same family (LMS or RLS) and also for the combination of one LMS and one RLS filter. These results are summarized in Table III, and constitute the basis for the study that we carry out in the rest of this section.

TABLE III

STEADY-STATE CROSS-EMSE FOR THE LMS AND RLS ADAPTIVE FILTER FAMILIES.

Combination type	steady-state cross-EMSE, $\zeta_{12}(\infty)$
μ_1-LMS and μ_2-LMS	$\dfrac{\mu_1\mu_2\sigma_v^2\text{Tr}\{\mathbf{R}\} + \text{Tr}\{\mathbf{Q}\}}{\mu_1 + \mu_2}$
β_1-RLS and β_2-RLS	$\dfrac{\beta_1\beta_2\sigma_v^2 M + \text{Tr}\{\mathbf{QR}\}}{\beta_1 + \beta_2}$
μ-LMS and β-RLS	$\mu\beta\sigma_v^2\text{Tr}\{(\mu\mathbf{R} + \beta\mathbf{I})^{-1}\mathbf{R}\} + \text{Tr}\{\mathbf{Q}(\mu\mathbf{R} + \beta\mathbf{I})^{-1}\mathbf{R}\}$

B. Combination of two LMS filters with different step sizes

A first example to illustrate the possibilities of combination schemes consists in putting together two filters with different adaptation speeds to obtain a new filter with improved tracking capabilities. Here, we illustrate the idea with a combination of two LMS filters with different step sizes, as was done in [12], but the idea can be straightforwardly generalized to other adaptive filter families (see, among others, [15], [19], [21], [23], [25], [30]).

Using the tracking model given by (11), in this section we analyze the theoretical steady-state EMSE for varying $\text{Tr}\{\mathbf{Q}\}$. In the studied scenario, the product $\sigma_v^2\text{Tr}\{\mathbf{R}\}$ remains fixed and is equal to 10^{-2}. To
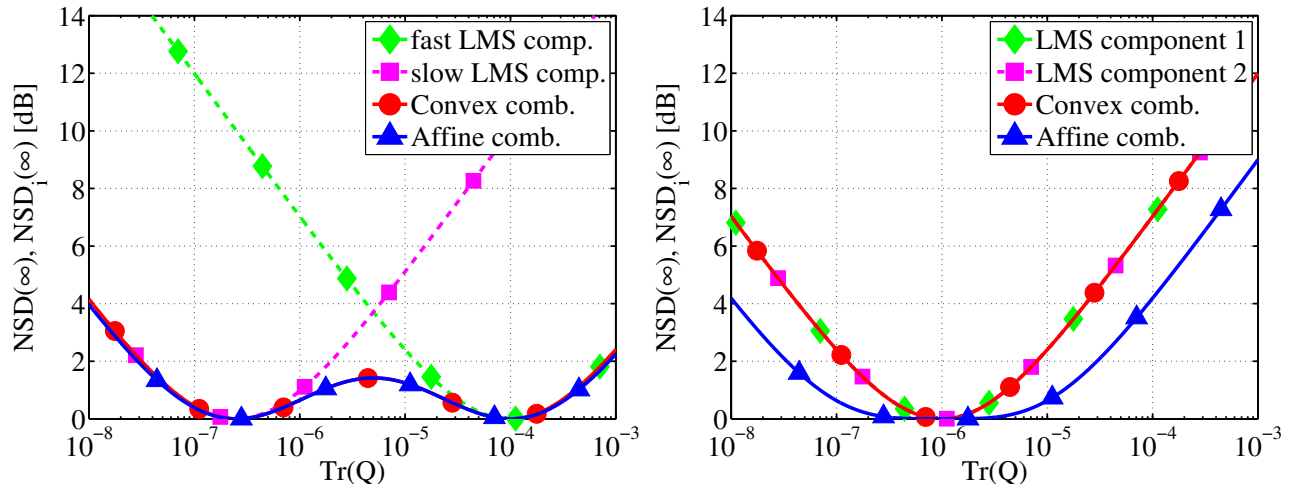
Fig. 2.   Steady-state NSD of a combination of two adaptive filters with different step sizes as a function of the speed of changes of the optimum solution. The figures illustrate the NSD incurred by the individual LMS filters, and by their optimum combination. LMS step sizes are: (left) $\mu_1 = 0.1$ and $\mu_2 = 0.005$; (right) $\mu_1 = 0.01$ and $\mu_2 = 0.010001$.

facilitate the comparison among filters, we will recur to the Normalized Squared Deviation (NSD), which we define as the EMSE excess with respect to the optimum LMS filter for a particular $\text{Tr}\{\mathbf{Q}\}$, i.e.,

$$\text{NSD}_i(\infty) = \frac{\zeta_i(\infty)}{\zeta_o(\infty)}, \ i = 1, 2, \quad \text{and} \quad \text{NSD}(\infty) = \frac{\zeta(\infty)}{\zeta_o(\infty)},$$

for the component LMS filters and their combination, respectively.

Figure 2 illustrates the theoretical NSD of individual LMS filters and combined schemes in two cases. The left panel considers LMS components with step sizes $\mu_1 = 0.1$ and $\mu_2 = \mu_1/20$. We can see that both individual filters become optimal (NSD = 0 dB) for a certain value of $\text{Tr}\{\mathbf{Q}\}$, and very rapidly increase their NSD level both for larger and smaller $\text{Tr}\{\mathbf{Q}\}$, i.e., for solutions that change relatively faster or slower than the speed of changes for which the filter is optimum. We can see that the NSD of each individual filter remains smaller than 2 dB for a range of about two orders of magnitude of $\text{Tr}\{\mathbf{Q}\}$. In contrast to this, we can check that the NSD of the combination of the two filters remains smaller than 2 dB when $\text{Tr}\{\mathbf{Q}\}$ changes by more than four orders of magnitude. It should be mentioned that, in this case, both the affine and convex combinations provide almost identical behavior.

A second possibility is illustrated in the right panel of Figure 2, for which the step sizes of the two LMS filters are set at almost identical values ($\mu_1 = 0.01$ and $\mu_2 = 0.010001$). In this case, both LMS components obtain approximately the same NSD for all values of $\text{Tr}\{\mathbf{Q}\}$. According to the theoretical results, it can be shown that in this example cases 1 and 2 enumerated in Table I are observed when $\text{Tr}(\mathbf{Q})$ is, respectively, smaller or larger than approximately 10^{-6}. The limitation in the range of $\lambda(n)$ makes the convex combination

perform exactly as any of the components, whereas the affine combination achieves an almost systematic gain of 3 dB with respect to the single filters [18], with the NSD remaining at values of less than 2 dB when $\text{Tr}\{\mathbf{Q}\}$ is changed by three orders of magnitude. As a result, in this case, the combination approach offers a possibility to improve the tracking capability of LMS filters. The examples in this section illustrate the following points:

- By employing a combination structure, we can improve the tracking ability of the adaptive implementation. This is important because the actual speed of change of the solution is rarely known *a priori*, and can even change over time. Later in Subsection IV-C we compare the combination framework with variable step-size (VSS) schemes, and show that combinations are more robust than VSS.

- When combining two LMS filters with different step sizes, the resulting combination cannot outperform the optimum filter for a given $\text{Tr}\{\mathbf{Q}\}$ [15]. Therefore, in this situation, if enough statistical information is *a priori* available, it would be preferable to use a single LMS filter with optimum step size. A different situation will be described in the next subsection.

C. Improving the tracking performance of LMS and RLS filters

When studying a combination of LMS and RLS filters following the same approach, an interesting result is obtained. Assume the tracking model (11) and consider the optimum LMS and RLS filters with adaptation parameters given by the expressions in Table II. As explained in [6], LMS will outperform RLS if $\mathbf{Q}$ is proportional to the autocorrelation matrix of the input signal, $\mathbf{R}$, and the opposite will occur when $\mathbf{Q} \propto \mathbf{R}^{-1}$. To illustrate this behavior, let us consider a synthetic example where $\mathbf{Q}$ is a mixture of $\mathbf{R}$ and $\mathbf{R}^{-1}$:

$$\mathbf{Q} = 10^{-5} \left[\alpha \frac{\mathbf{R}}{\text{Tr}\{\mathbf{R}\}} + (1 - \alpha) \frac{\mathbf{R}^{-1}}{\text{Tr}\{\mathbf{R}^{-1}\}} \right], \tag{12}$$

with $\alpha \in (0, 1)$.

Figure 3 plots the steady-state optimum EMSE that can be achieved by both types of filters with optimum settings (μ_{o}-LMS and β_{o}-RLS), when $\mathbf{Q}$ smoothly changes between $\mathbf{R}^{-1}$ (for $\alpha = 0$) and $\mathbf{R}$ (for $\alpha = 1$). Other settings for this scenario are: optimal solution and filters length $M = 7$, noise variance $\sigma_v^2 = 10^{-2}$, and $\mathbf{R}$ is a Toeplitz matrix whose first row is given by $\frac{1}{7}[1, 0.8, 0.8^2, \cdots, 0.8^6]$. According to (12), the value of the trace of $\mathbf{Q}$ remains constant when changing the value of α, meaning that the EMSE of the optimum LMS is approximately equal to -35 dB for any value of α. If a non-optimum step size is selected, the LMS filter will incur a larger EMSE. Thus, the region of feasible EMSE values for this family of filters is given by the area above the -35 dB level in Fig. 3. Similarly, the region above the curve for the optimum RLS performance represents all feasible EMSE values for the family of RLS filters. We can see that, depending on the value of α, the optimal EMSE for RLS filters can be larger or smaller than the optimal EMSE for LMS filters. Finally, the dark green area in Fig. 3 is a feasible region for both LMS and RLS filters.
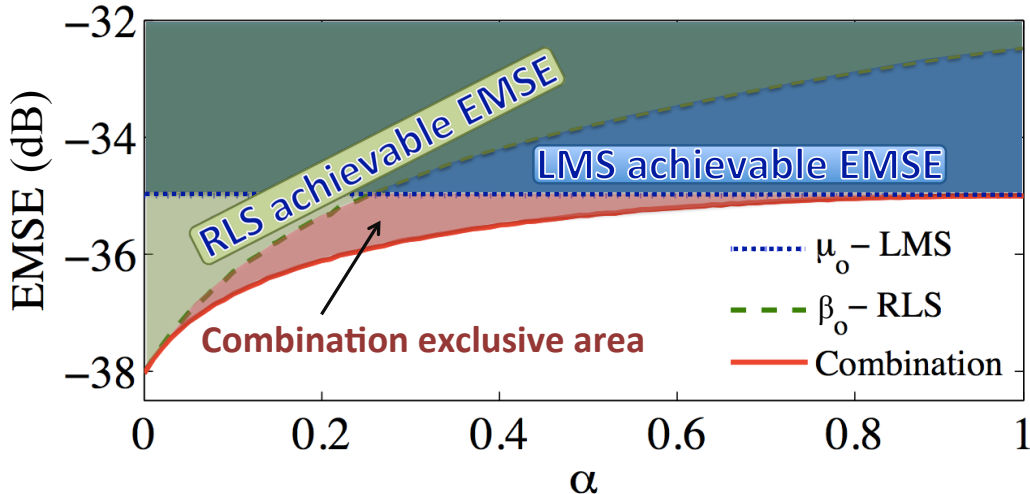
Fig. 3. Tracking performance of a combination of LMS and RLS filters when $\mathbf{Q}$ smoothly changes between $\mathbf{R}^{-1}$ (for $\alpha = 0$) and $\mathbf{R}$ (for $\alpha = 1$). The performance of optimally adjusted filters is depicted with curves. The blue and light green regions represent EMSEs that can be obtained with LMS and RLS filters, respectively, whereas the dark green area comprises feasible EMSE values for both LMS and RLS filters. Finally, the red region contains EMSE values that can be obtained with combinations of LMS and RLS (but not with these filters individually).

The performance that can be achieved using a combination of one LMS and one RLS filters is also illustrated. The red curve represents the steady-state tracking EMSE of the combination when using μ_{o} and β_{o} for the LMS and RLS filters in the combination. As expected, the optimal EMSE for the combination is never larger than the smallest of the filter component EMSEs. More importantly, in this case we can check the existence of an area of EMSE values which is exclusive to the combination scheme. In other words, the red area in Figure 3 is a feasible region for the operation of the combined scheme, but not for single LMS or RLS filters. It can be shown that, in this case, the optimum mixing parameter of the combination lies in interval $(0, 1)$, so this conclusion is valid for convex combination schemes [36]. This is a useful conclusion that can be used to improve the tracking performance beyond the limits of individual filters.

IV. ESTIMATING THE COMBINATION PARAMETER

Since the optimum linear combiner (8) is unrealizable, many practical algorithms have been proposed in the literature to adjust the mixing parameter in convex [12], [37], [38] and affine combinations [16]–[18], [39], [40]. Rewriting (2) as

$$y(n) = y_2(n) + \lambda(n)[y_1(n) - y_2(n)], \tag{13}$$

we can reinterpret the adaptation of $\lambda(n)$ as a "second layer" adaptive filter of length one, so that in principle any adaptive rule can be used for adjusting the mixing parameter. However, this filtering problem has some

particularities, namely, the strong and time-varying correlation between $y_1(n)$ and $y_2(n)$. This implies that the power of the difference signal, $[y_1(n) - y_2(n)]$, is also time-varying depending, e.g., on the signal-to-noise conditions and on whether the individual filters are operating in convergence phase or steady state, etc.

Using a stochastic gradient search to minimize the quadratic error $e^2(n)$ of the overall filter, defined in (4), reference [16] proposed the following adaptation for the mixing parameter in an affine combination (i.e., without imposing restrictions on $\lambda(n)$),

$$\lambda(n+1) = \lambda(n) - \frac{\mu_\lambda}{2} \frac{\partial e^2(n)}{\partial \lambda(n)} = \lambda(n) + \mu_\lambda e(n)[y_1(n) - y_2(n)]. \tag{14}$$

We will refer to this case as *aff-LMS* adaptation, with μ_λ being a step-size parameter. As discussed in [16], a large step size should be used in this rule to ensure an adequate behavior of the affine combination. However, this can cause instability during the initial convergence of the algorithm, which was circumvented in [16] by constraining $\lambda(n)$ to be less than or equal to one. Even with this constraint, the combination scheme may stay away from the optimum EMSE (see, e.g., [18]). This is a direct consequence of the time-varying power of $[y_1(n) - y_2(n)]$, which makes selection of μ_λ a difficult task.

In order to obtain a more robust scheme, it is possible to recur to normalized adaptation schemes. Using a rough (low-pass filtered) estimation of the power of $[y_1(n) - y_2(n)]$, a power normalized version of *aff-LMS* was proposed in [18], and is summarized in Table IV. This *aff-PN-LMS* algorithm does not impose any constraints on $\lambda(n)$, is less sensitive to the filtering scenario, and converges in the mean-square sense if the step size is selected in the interval $0 < \mu_\lambda < 2$.

Rather than directly adjust $\lambda(n)$ as in the affine case, convex combination schemes recur to activation functions to keep the mixing parameter in the range of interest. For example, reference [12] proposed an adaptation scheme for an auxiliary parameter $a(n)$ that is related to $\lambda(n)$ via the *sigmoid* function

$$\lambda(n) = \text{sgm}[a(n)] = \frac{1}{1 + e^{-a(n)}}. \tag{15}$$

Recurring to this activation function (or similar ones), $a(n)$ can be adapted without constraints, and $\lambda(n)$ will be automatically kept inside the interval $(0, 1)$ at all times. Using a gradient descent method to minimize the quadratic error of the overall filter, $e^2(n)$, two algorithms were proposed to update $a(n)$: the *cvx-LMS* algorithm [12] and its power normalized version, *cvx-PN-LMS* [37], whose update equations are given by

$$a(n+1) = a(n) + \mu_a \lambda(n)[1 - \lambda(n)]e(n)[y_1(n) - y_2(n)], \qquad \textit{(cvx-LMS)} \tag{16}$$

$$a(n+1) = a(n) + \frac{\mu_a}{p(n)} \lambda(n)[1 - \lambda(n)]e(n)[y_1(n) - y_2(n)]. \qquad \textit{(cvx-PN-LMS)} \tag{17}$$

TABLE IV

POWER NORMALIZED ADAPTATION OF $\lambda(n)$ IN AFFINE COMBINATIONS AND OF AUXILIARY PARAMETER $a(n)$ IN CONVEX COMBINATIONS. UNNORMALIZED RULES *aff-LMS* AND *cvx-LMS* ARE OBTAINED SETTING $p(n) = 1$.

Algorithm	Update Equations
aff-PN-LMS [18]	$\lambda(n+1) = \lambda(n) + \dfrac{\mu_\lambda}{\epsilon + p(n)} e(n)[y_1(n) - y_2(n)]$
	$p(n) = \eta\, p(n-1) + (1-\eta)[y_1(n) - y_2(n)]^2$
	$0 \ll \eta < 1; \quad \epsilon > 0$ is a small constant
cvx-PN-LMS [37]	$a(n+1) = a(n) + \dfrac{\mu_a}{p(n)} \lambda(n)[1 - \lambda(n)]e(n)[y_1(n) - y_2(n)]$
	$p(n) = \eta\, p(n-1) + (1-\eta)[y_1(n) - y_2(n)]^2$
	$0 \ll \eta < 1; \quad$ Constraint: $-a^+ \le a(n+1) \le a^+$

Here, $p(n)$ is a low-pass filtered estimation of the power of $[y_1(n) - y_2(n)]$. These algorithms are also shown in Table IV for further reference. Compared to the *cvx-LMS* scheme, *cvx-PN-LMS* is more robust to signal-to-noise ratio (SNR)[2] changes, and simplifies the selection of step size μ_a [37].

The factor $\lambda(n)[1 - \lambda(n)]$, that appears in (17), reduces the gradient noise when $\lambda(n)$ gets too close to the values of zero or one. In this situation, adaptation would virtually stop if $a(n)$ were allowed to grow unchecked. To avoid this problem, the auxiliary parameter $a(n)$ is restricted by saturation to a symmetric interval $[-a^+,\ a^+]$, which ensures a minimum level of adaptation[3] [12], [39]. This truncation procedure constrains $\lambda(n)$ to the interval $\text{sgm}[-a^+] \le \lambda(n) \le \text{sgm}[a^+]$. In order to allow $\lambda(n)$ to take a value equal to zero or one, a scaled and shifted version of the sigmoid was proposed in [38],

$$\lambda(n) = \varphi[a(n)] = \frac{\text{sgm}[a(n)] - \text{sgm}[-a^+]}{\text{sgm}[a^+] - \text{sgm}[-a^+]}, \tag{18}$$

which ensures that $\lambda(n)$ attains values 1 and 0 for $a(n) = a^+$ and $a(n) = -a^+$, respectively. Using (18), the *cvx-PN-LMS* rule becomes

$$a(n+1) = a(n) + \frac{\mu_a}{p(n)} e(n)[y_1(n) - y_2(n)]\text{sgm}[a(n)]\left\{1 - \text{sgm}[a(n)]\right\} \Big|_{-a^+}^{a^+}. \tag{19}$$

As we can see, $a(n)$ is truncated at every iteration to keep it inside the interval $[-a^+,\ a^+]$.

When deciding between affine or convex combinations with the above gradient-based rules, one should be aware of the following facts:

- Optimally adjusted affine combinations attain smaller EMSE than convex ones in some situations (cases 1 and 2 of Table I).

---

[2]According to the linear regression model (1), the SNR is defined as $\text{SNR} = [\mathbf{w}_o^\top(n)\mathbf{R}\mathbf{w}_o(n)]/\sigma_v^2$.

[3]A common choice in the literature is $a^+ = 4$.

- Due to the constraints on the value of $\lambda(n)$, *cvx-LMS* and *cvx-PN-LMS* do not diverge, and large values of the step size μ_a can be used.

- Gradient adaptation of the combination parameter implies that combinations introduce some additional gradient noise, which should be minimized with an adequate selection of μ_a or μ_λ. In this sense, the factor $\lambda(n)[1 - \lambda(n)]$ that appears in the adaptation of $a(n)$ with *cvx-PN-LMS* reduces the gradient noise when the mixing parameter becomes close to 0 or 1.

In our experience, the last two issues have an important impact on the performance of the combination, to the extent that convex combinations may actually be preferred over affine ones, unless in situations where significant EMSE gains can be anticipated for the affine combination from the theoretical analysis. The next subsection will compare the performance of these rules through simulations.

Before concluding the section, we should point out that other rules can be found in the literature for the adaptation of the mixing parameter, such as the *least squares* rule from [39], the sign algorithm proposed in [41], or a method relying on the ratio between the estimated errors of the filter components [16]. Nevertheless, in the following we will restrict our discussion to the methods that have been presented in this section, which are the most frequently used in the literature.

A. Convergence properties of combination filters

To examine the convergence properties of *aff-PN-LMS* and *cvx-PN-LMS*, we consider the combination of two normalized LMS (NLMS) filters with step sizes $\mu_1 = 0.5$ and $\mu_2 = 0.01$. The optimum solution is a stationary vector of length 7, the covariance matrix of the input signal is $\mathbf{R} = \frac{1}{7}\mathbf{I}$, and the variance of the observation noise is adjusted to get an SNR of 20 dB. Different step sizes have been explored for the combination: $\mu_a = [0.25, 0.5, 1]$ for *cvx-PN-LMS*, while $\mu_\lambda = \mu_a/800$ is used for *aff-PN-LMS* to get comparable steady-state error. Regarding these step-size values, we can see that the range of practical values for μ_a is within the usual range of steps sizes used with normalized schemes, whereas for the affine combination much smaller values are required for comparable performance. This fact simplifies the selection of the step size in the convex case.

Figure 4 illustrates the performance of affine and convex combinations averaged over 1000 experiments. Subfigures (b)-(d) compare the convergence of both schemes with respect to the optimum selection of the mixing parameter given by (8). In all cases, the combination schemes converge first to the EMSE level of the fast filter (-30 dB), and after a while follow the slow component to get a final EMSE of around -50 dB. It is interesting to see that *cvx-PN-LMS* shows near optimum selection of the mixing parameter for all three values of μ_a, while the affine combination may incur a significant delay, especially for the smallest μ_λ. Subfigure 4-(a) plots the excess steady-state error of both schemes with respect to $\zeta_2(\infty)$ as a function

(a)

(b) $\mu_a = 0.25$

(c) $\mu_a = 0.5$

(d) $\mu_a = 1$

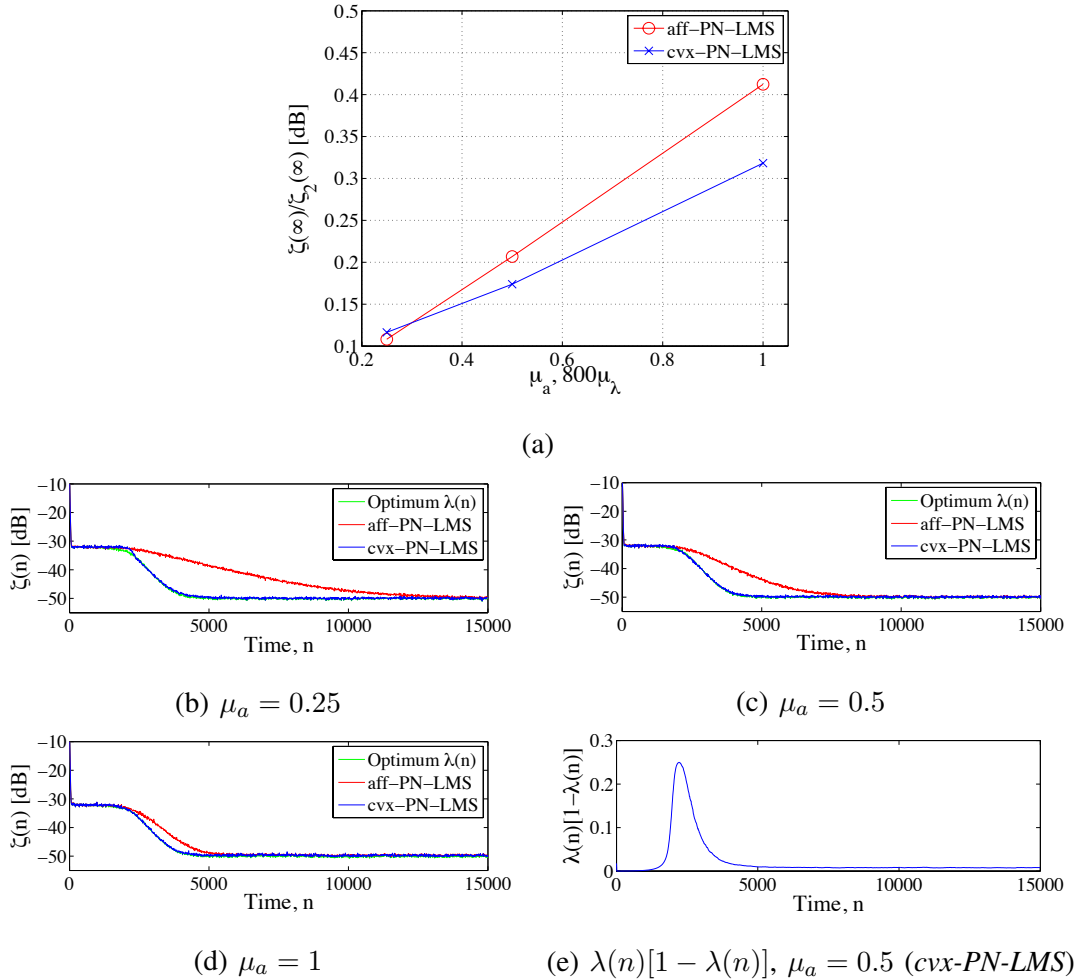(e) $\lambda(n)[1 - \lambda(n)]$, $\mu_a = 0.5$ (*cvx-PN-LMS*)

Fig. 4.   Performance of *aff-PN-LMS* and *cvx-PN-LMS* when combining two NLMS adaptive filters with step sizes $\mu_1 = 0.5$ and $\mu_2 = 0.01$. (a) Steady-state excess error with respect to the slow filter component for three different values of the step sizes. Subfigures (b), (c), and (d) compare the convergence of the affine and convex rules (for three values of μ_a and $\mu_\lambda = \mu_a/800$) with the combination scheme using the optimum mixing parameter. (e) Evolution of factor $\lambda(n)[1 - \lambda(n)]$ for *cvx-PN-LMS*.

of the step size, and shows that this faster convergence of *cvx-PN-LMS* with respect to *aff-PN-LMS* is not in exchange of larger residual error.

The fact that *cvx-PN-LMS* has the ability to switch rapidly between the fast and slow filter components while at the same time minimizing the residual error in steady state is due to the incorporation of the activation function, whose derivative propagates to the update rule. In other words, we can view the effective step size of *cvx-PN-LMS* as being $\mu_a \lambda(n)[1 - \lambda(n)]$ (see Table IV), and the evolution of the multiplicative factor, represented in Figure 4-(e), shows that this effective step size becomes large when the combination needs to switch between filter components, while becoming small in steady state, thus minimizing the residual error after convergence is complete.

B. Benefits of power normalized updating rules

In order to illustrate the benefits of power normalized updating rules, we compare the behavior of a convex combination of two NLMS filters with step sizes $\mu_1 = 0.5$ and $\mu_2 = 0.01$, employing both the *cvx-LMS* rule and its power normalized version (*cvx-PN-LMS*) for the combination layer. The optimum solution is a length-30 nonstationary vector, which varies according to the random-walk model given by (11). The covariance matrices of the change of the optimum solution and of the input signal are given respectively by $\mathbf{Q} = \sigma_q^2 \mathbf{I}$ and $\mathbf{R} = \frac{1}{30}\mathbf{I}$. The variance of the observation noise is adjusted to get different SNR levels. The step size for the *cxv-LMS* rule has been set to $\mu_a = 1000$ and for its power normalized version (*cvx-PN-LMS*), we have set $\mu_a = 1$.

Figure 5 shows the steady-state NSD of the individual filters and of their convex combinations obtained with the *cxv-LMS* rule and with the *cvx-PN-LMS* scheme, as a function of the speed of changes of the optimum solution [$\mathrm{Tr}(\mathbf{Q}) = 30\sigma_q^2$]. The left panel considers an SNR of 5 dB and the right panel, an SNR of 30 dB. We can observe that the combination scheme with the *cxv-LMS* rule results in a suboptimal performance when the optimum solution changes very fast [for large $\mathrm{Tr}(\mathbf{Q})$]. This is due to the fact that both component filters are incurring a very significant error, resulting in a non-negligible gradient noise when updating the auxiliary parameter $a(n)$ with *cxv-LMS* [37]. The performance of *cxv-LMS* also degrades, whatever the value of $\mathrm{Tr}(\mathbf{Q})$, as the SNR decreases. On the other hand, when the *cvx-PN-LMS* rule is employed, the combination shows a very stable operation, and behaves as well as the best component filter not only for any SNR, but also for all values of $\mathrm{Tr}(\mathbf{Q})$. A similar behavior is also observed when we compare the *aff-LMS* rule to its power normalized version (*aff-PN-LMS*) to update $\lambda(n)$ in affine combinations [18].

C. Combination schemes versus VSS LMS

We have illustrated so far how combination schemes can be exploited to provide filters with improved performance, relying on adaptive rules to learn the most adequate value for the mixing parameter. A reasonable question is whether the same improvement could be achieved by an individual but more complex filter structure. Although the answer to this question is generally positive, these more complex solutions normally require further statistical knowledge about the filtering scenario that is normally unavailable, while combination approaches offer a more versatile and robust approach. To illustrate this point, we will examine again the identification of a time-varying solution.

Adaptive filters usually have two conflicting goals: on one hand, to track variations of the parameter vector they are trying to estimate and, on the other hand, to suppress measurement noise and errors due to undermodeling. The first goal requires a fast filter, one that quickly corrects any mismatch between the input and the estimates, while the second goal would benefit from a slower filter that averages out measurement
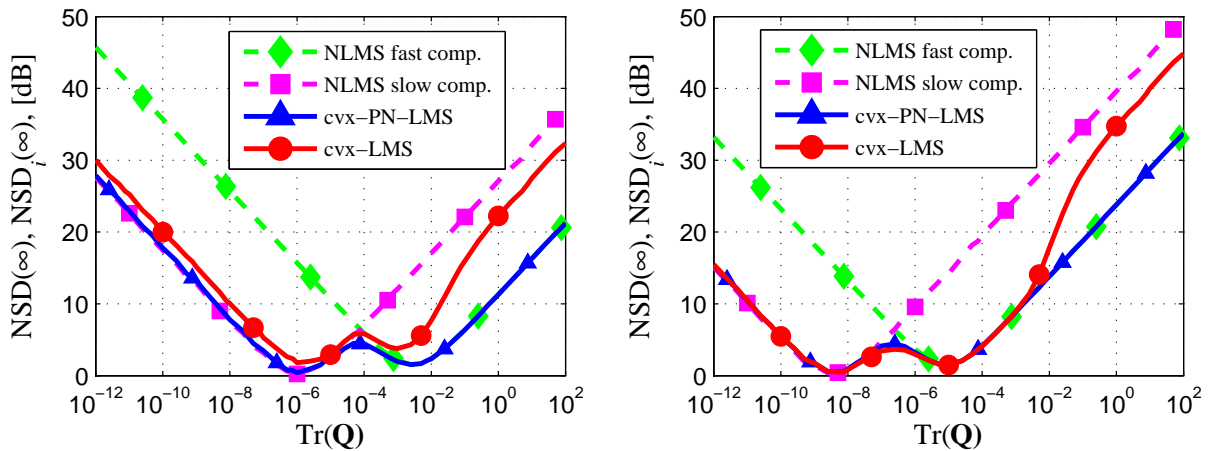
Fig. 5. Steady-state NSD of a convex combination of two NLMS filters with different step sizes as a function of the speed of changes of the optimum solution. The figures illustrate the NSD incurred by the individual NLMS filters and by their convex combinations. NLMS step sizes are $\mu_1 = 0.5$ and $\mu_2 = 0.01$; combination rules: *cvx-LMS* ($\mu_a = 1000$) and *cvx-PN-LMS* ($\mu_a = 1$, $\eta = 0.9$); signal-to-noise ratios: (left) SNR = 5 dB and (right) SNR = 30 dB. All results were averaged over 50000 iterations once the algorithms reached steady state and over 50 independent runs.

noise. We can emphasize one goal over the other by the choice of a design variable such as the step size in LMS or the forgetting factor in RLS. However, in a nonstationary environment, the optimum choice of the step size or the forgetting factor will continuously change, which has motivated the proposal of different methods to update the step size (or forgetting factor) —see e.g., [42]–[46]. Most existing variable step size (VSS) algorithms implement procedures that rely on the available input and error signals. However, in the absence of additional information (e.g., background noise level, whether the filter is still converging or has reached steady state), these algorithms can fail to identify the most convenient step size, especially if the filtering conditions change over time.

As an alternative to the VSS algorithms, combinations of adaptive filters with different step sizes (or forgetting factors) can be particularly useful. One advantage of this approach is that it performs reasonably well for large variations in input signal-to-noise ratio and speed of change of the true parameter vector. Figure 6 compares the performance, in an identification problem, of the robust variable step-size methods proposed in [45] and [46] against the performance of convex combination of two LMS adaptive filters with different step sizes. The fast filter uses $\mu_1 = 0.01$ and the slow filter, $\mu_2 = 0.001$. The figure shows the mean-square deviation (MSD) of a 10-tap filter in a tracking situation like the one described by (11), with $\mathbf{Q} = \sigma_q^2 \mathbf{I}$, for time-varying σ_q^2 as indicated in the figure. In order to provide a fair comparison, the parameters for the VSS methods were optimized for step sizes in the range $[\mu_2, \mu_1]$, and the figure shows the performance of the three methods for five different values of the optimum step size μ_o for a single LMS
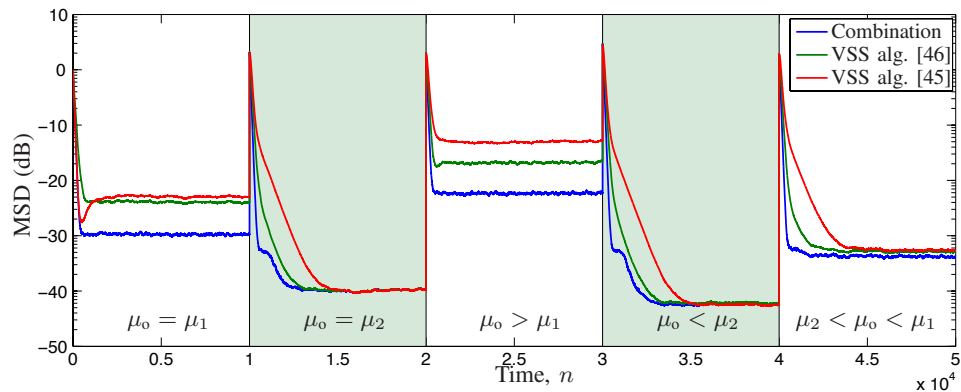
Fig. 6.   Mean-square deviation (MSD) obtained by VSS algorithms and by the convex combination of two LMS filters with $\mu_1 = 0.01$ and $\mu_2 = 0.001$. The value of σ_q^2 changes with time, and the corresponding optimum LMS step sizes (μ_o) are indicated in the figure.

filter: from less than μ_2 to more than μ_1. It is seen that the convex combination scheme is able to deliver the best possible performance in all conditions. In other words, it shows a more robust performance given the lack of knowledge about the true value of σ_q^2, and attains an overall performance that could not be achieved by a standard LMS filter or by the VSS schemes.

V. COMMUNICATION BETWEEN COMPONENT FILTERS

One simple way to improve the performance of combination schemes is to allow some communication between the component filters. Since the component filters are usually designed to perform better in different environmental conditions, information from the best filter in a given set of conditions can be used to boost the performance of the other filters. In some situations, the gain for the overall output can be significant.

We describe here two approaches for inter-filter communication: the weight transfer scheme of [47], [48], and the weight feedback of [49]. Both are applicable when the component filters have the same length. The difference between these methods is that the first allows only communication from one of the filters to the other, whereas the second approach feeds back the overall combined weights to the component filters, as depicted in Figure 7.

Consider for example the combination of two component filters, one designed for a fast-changing environment ("fast" filter), and the other for a slow-changing environment ("slow" filter). In this situation, during the initial convergence, or after an abrupt change in the optimum weight vector $\mathbf{w}_o(n)$, the slow filter may lag behind the fast filter (see Figure 8). After the fast filter reaches the steady state, the combination would
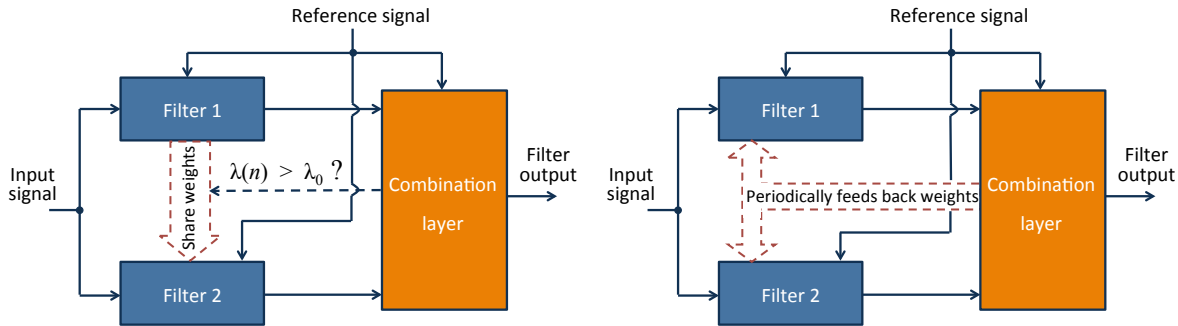
Fig. 7. Communication between sub-filters. Left panel: weight transfer; right panel: weight feedback.

TABLE V

INTER-FILTER COMMUNICATION SCHEMES (EXEMPLIFIED FOR COMBINATIONS OF TWO FILTERS).

	Gradual transfer [47]	Simple copy [48]	Feedback [49]
Filter communication	$\mathbf{w}_2(n) \leftarrow \ell\mathbf{w}_2(n) + (1-\ell)\mathbf{w}_1(n)$	$\mathbf{w}_2(n) \leftarrow \mathbf{w}_1(n)$	$\mathbf{w}_1(n), \mathbf{w}_2(n) \leftarrow \mathbf{w}(n)$
Condition	$\lambda(n) \geq \lambda_0$	$\lambda(n) \geq \lambda_0$ and $n = kN_0$	$n = kN_0$
Parameters	$0 < \ell < 1; \quad 0 \ll \lambda_0 < 1$	$N_0 \geq 2; \quad 0 \ll \lambda_0 < 1$	$N_0 \geq 2$

need to wait until the slow filter catches up. This can be avoided by either leaking [47] or copying [48] the fast filter coefficients to the slow filter at the appropriate times.

The good news is that the mixing parameter is already available to help decide when to transfer the coefficients —we only need to check if $\lambda(n)$ is close to one (so that the overall output is essentially the fast filter). Therefore, the idea is to choose a threshold λ_0, and modify the slow filter update whenever $\lambda(n) \geq \lambda_0$, as shown in Table V. The two methods are similar: in the gradual transfer scheme of [47], whenever $\lambda(n)$ is larger than the threshold, we allow the fast filter coefficients to gradually "leak" into the slow filter. The other method simply copies the fast filter coefficients to the slow filter, and thus has a smaller computational cost; however, the condition for transfer has to be modified. If we allowed the fast filter coefficients to be copied to the slow filter whenever $\lambda(n) \geq \lambda_0$, the combination would be stuck forever at the fast filter (the slow filter adaptation would be irrelevant). For this reason the transfer is only allowed at periodic intervals of length $N_0 \geq 2$.

The weight feedback scheme, on the other hand, copies the combined weight vector $\mathbf{w}(n) = \lambda(n)\mathbf{w}_1(n) + (1-\lambda)\mathbf{w}_2(n)$ periodically to $\mathbf{w}_1(n)$ and $\mathbf{w}_2(n)$, as shown in Figure 7 and Table V. This method is simpler to set up, since only one parameter needs to be tuned, but requires the explicit computation of $\mathbf{w}(n)$ at
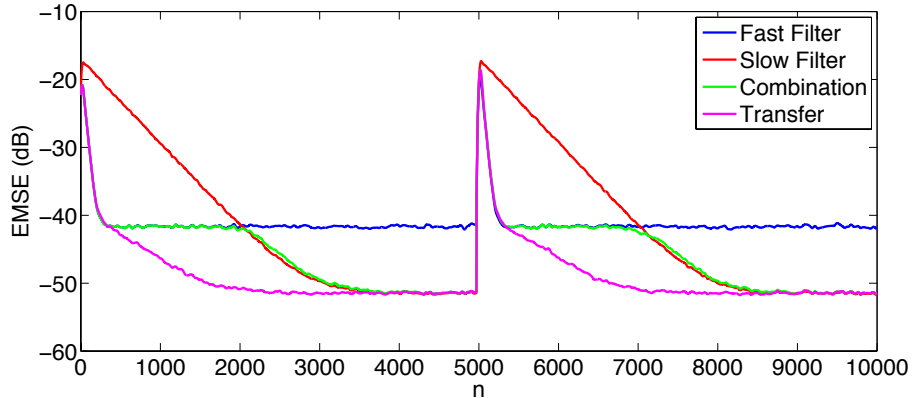
Fig. 8.   Performance of weight transfer schemes. The plots show the EMSE obtained with the component filters and standard convex combination without weight transfer, and the second method of Table V, using $N_0 = 2$ and $\lambda_0 = 0.982$. The curves are averages of $L = 1500$ simulations. The components are two NLMS algorithms, one using $\mu_1 = 0.1$, the other using $\mu_2 = 0.01$.

intervals of N_0 samples. Since N_0 can be chosen reasonably large, the additional cost per sample is modest. Even though these methods introduce additional parameters that need to be fixed, their selection is not very problematic and the transfer methods show a behavior rather robust with respect to them (a detailed discussion is not included here, but the interested readers are referred to [47]–[49]).

Figure 8 provides an example of the operation of the simple-copy method for length $M = 7$ filters (the other methods have similar performance). The value of $\mathbf{w}_o(n)$ changes abruptly at the middle of the simulation ($n = 5 \times 10^4$).

VI. COMBINATION OF SEVERAL ADAPTIVE FILTERS

Up to now, we have considered combinations of just two adaptive filters. However, combining K adaptive filters (with $K > 2$) makes it possible to further increase the robustness and versatility of combination schemes. The combination of an arbitrary number of filters can be mainly used to:

- Simplify the selection of a parameter. For instance, paying attention to the tracking scenario depicted in the left plot of Fig. 2, the robustness of the scheme would be increased for $\text{Tr}\{\mathbf{Q}\} > 10^{-3}$ if a third filter with step size equal to one is incorporated to the combination [50].

- Alleviate several compromises simultaneously. For instance, regarding the selection of the step size, μ, and the length of an adaptive filter, M, we can combine four adaptive filters with settings $\{\mu_1, M_1\}$, $\{\mu_2, M_1\}$, $\{\mu_1, M_2\}$ and $\{\mu_2, M_2\}$. Another example was proposed in [25] and it is included in Section XI, where a combination of several filters (linear and nonlinear) is designed to alleviate simultaneously the compromise related with the step size selection, and with the presence or absence of nonlinearities in the filtering scenario.
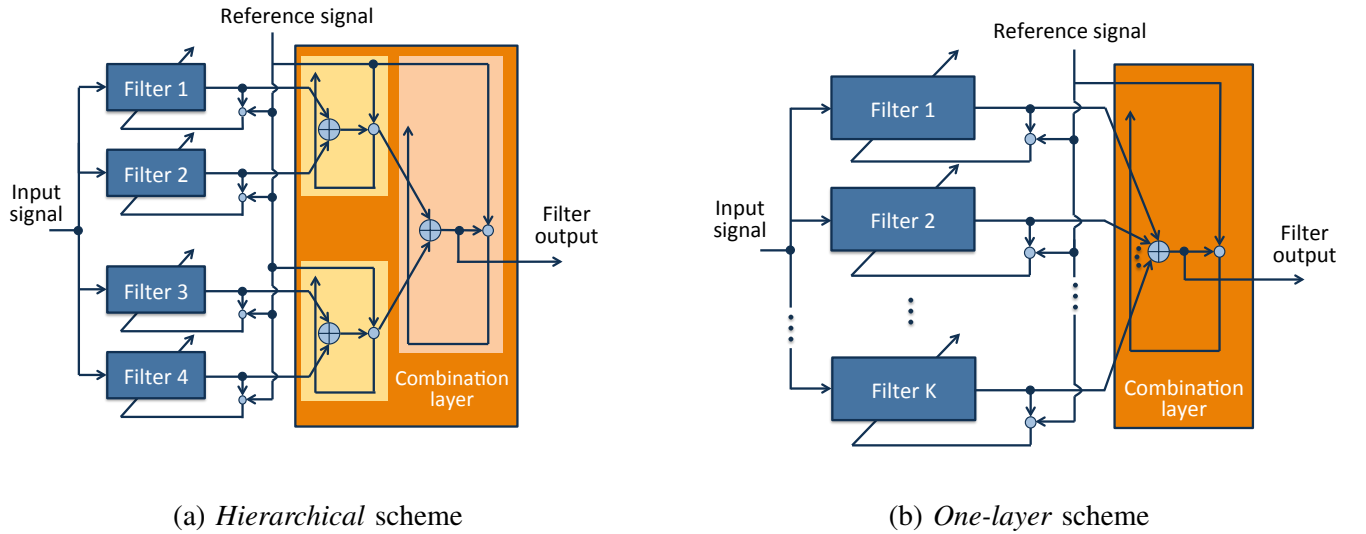
(a) *Hierarchical* scheme   (b) *One-layer* scheme

Fig. 9.   Hierarchical and one-layer combination schemes for an arbitrary number of filters K. (a) Hierarchical scheme: In the first layer, we carry out two combinations of two adaptive filters with complementary settings, and their outputs are combined in the upper layer. (b) Combination of K adaptive filters following the one-layer approach. The affine schemes of [50] and [19] adapt $K-1$ mixing parameters, whereas the convex combination of [19], [51] requires the adaptation of K parameters.

In the literature, two different approaches for the combination of several adaptive filters have been proposed, both for affine and convex approaches. These schemes differ in the topology employed to perform the combination as described below.

The hierarchical scheme

This approach combines K adaptive filters employing different layers, where only combinations of two elements are considered at a time. For instance, the output of a hierarchical combination of four filters depicted in Fig. 9(a) reads:

$$y(n) = \lambda_{21}(n)\{\lambda_{11}(n)y_1(n) + [1-\lambda_{11}(n)]y_2(n)\} + [1-\lambda_{21}(n)]\{\lambda_{12}(n)y_3(n) + [1-\lambda_{12}(n)]y_4(n)\}, \quad (20)$$

where $\lambda_{ij}(n)$ refers to the mixing parameter of the j-th combination in the i-th layer. All mixing parameters are adapted in order to minimize the power of the local combined error. For their update, we can follow the same adaptive rules (convex or affine) reviewed previously for the case of a combination of two filters.

The one-layer scheme

We can combine an arbitrary number of filters employing an alternative approach based on one-layer combination as depicted in Fig. 9(b). Focusing on affine combinations of M adaptive filters with outputs

$y_k(n)$, with $k = 1, ..., K$, references [19] and [50] proposed two one-layer combination schemes whose output is given by

$$y(n) = \sum_{k=1}^{K-1} \lambda_k(n)y_k(n) + \left[1 - \sum_{k=1}^{K-1} \lambda_k(n)\right] y_K(n). \tag{21}$$

Different adaptive rules were proposed in the literature to update $\lambda_k(n)$, with $k = 1, ..., K - 1$, following, for instance, LMS or RLS approaches [19], or estimating the $K - 1$ affine mixing parameters as the solution of a least-squares problem [50]. The incorporation of convex combination constraints forces the inclusion of an additional mechanism (similar to the sigmoidal activation) to make all mixing parameters remain positive and add up to one. This scheme was proposed by [51] and [19] as an extension to the standard convex combination of two adaptive filters, and obtains the combined output as

$$y(n) = \sum_{k=1}^{K} \lambda_k(n)y_k(n). \tag{22}$$

As in the case of combining two filters, instead of adapting directly the K mixing parameters, K auxiliary parameters $a_k(n)$ are updated following a gradient descent algorithm. The relation between $\lambda_k(n)$ and $a_k(n)$ is based on the *softmax* activation function

$$\lambda_k(n) = \frac{\exp[a_k(n)]}{\sum_{j=1}^{K} \exp[a_j(n)]}, \quad \text{for } k = 1, ..., K. \tag{23}$$

This activation function is a natural extension of the *sigmoid* used in the binary case to map several real parameters to a probability distribution [52], as required by a convex combination, where all parameters must remain positive and sum up to one.

Although both multifilter structures can improve the performance beyond the combination of just two filters, one useful characteristic of hierarchical schemes is its ability to extract more information about the filtering scenario from the evolution of the mixing parameters, since each combination usually combines two adaptive filters that only differ in the value of a setting (step size, length, etc.), and the combination parameter selects the best of both competing models.

VII. REDUCED COST COMBINATIONS

Combination schemes require running two or more filters in parallel, which may be a concern in applications in which computational cost is at a premium. In many situations, however, the additional computational cost of adding one or more filters can be made just slightly higher than the cost of running a single filter. In this section we describe a few methods to reduce the cost of combination schemes.

Use a low-cost filter as companion to a high-cost one

Although straightforward, several useful results fall in this class. For example, consider the previously mentioned case of a combination of an RLS and an LMS filter. This structure enhances the tracking performance of a single RLS filter, and only requires a modest increase in computational complexity. A lattice implementation of RLS has a computational cost of about $16M$ multiplications, $8M$ additions, and $8M$ divisions [6]. Compared with the approximately $2M$ additions and $2M$ multiplications required by LMS, the combination scheme will increase the cost by only about 10% over that of a single RLS. For other stable implementations of RLS, such as QR methods, the computational cost is even higher, so the burden of adding an LMS component would be almost negligible[4].

Other situations in which it is useful to combine a low-cost and a higher-cost filter are: the combination of a simple linear filter with a Volterra nonlinear filter, as described in Section XI; and the use of a short filter combined with a longer filter, as described for example in [53]. The short filter is responsible for tracking fast variations in the weight vector, whereas the long filter is responsible for guaranteeing a small bias in times of slower variation of the optimum weight vector.

Parallelization

Since the component filters are running independently (apart from occasional information exchanges as in Section V), combination schemes are easy to implement in parallel form. In an FPGA implementation, for example, this means that implementing a combination of filters will not require a faster clock rate, compared to running a single filter. Depending on the hardware in which the filters are to be implemented, this is an important advantage.

Take advantage of redundancies in the component filters

This approach can provide substantial gains in some situations, but depends on the kind of filters that are being combined. For example, transform-domain or subband filters can share the transform or filter bank blocks that process the input signal $\mathbf{u}(n)$.

A second example is based on the following observation [54]: if filters using the same input regressor $\mathbf{u}(n)$ are combined, the weight estimates $\mathbf{w}_i(n)$ will be, most of the time, similar. Therefore, one could update only one of the filters (say, $\mathbf{w}_1(n)$), and the *difference* between this filter and the others. The important observation is that, if the entries of $\mathbf{w}_1(n)$ and $\mathbf{w}_2(n)$ are real values ranging from (say) -1 to $+1$, the range

---

[4]Note that, for their operation, combination schemes require just the outputs of component filters (see Section IV), so lattice or QR implementations of RLS can readily be incorporated into combination schemes without modifications.

TABLE VI

REDUCED-COST COMBINATION THROUGH DIFFERENCE FILTER (USING TWO LMS FILTERS AS EXAMPLE)

Algorithm Step	Equation	Wordlength
1 — First filter error:	$e_1(n) = d(n) - \mathbf{u}^\top(n)\mathbf{w}_1(n)$	Full
2 — Update first filter:	$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) + \mu_1 e_1(n)\mathbf{u}(n)$	Full
3 — Output of difference filter:	$\delta y_2(n) = \mathbf{u}^\top(n)\boldsymbol{\delta}\mathbf{w}_2(n)$	Reduced
4 — Second filter error:	$e_2(n) = e_1(n) + \delta y_2(n)$	Full
5 — Output error:	$\lambda(n)e_1(n) + (1 - \lambda(n))e_2(n)$	Full
6 — Update difference filter:	$\mathbf{e}_\delta(n) = \mu_2 e_2(n) - \mu_1 e_1(n)$	Reduced
	$\boldsymbol{\delta}\mathbf{w}_2(n+1) = \boldsymbol{\delta}\mathbf{w}_2(n) + e_\delta(n)\mathbf{u}(n)$	

of the entries of $\boldsymbol{\delta}\mathbf{w}_2(n) = \mathbf{w}_2(n) - \mathbf{w}_1(n)$ will most of the time be smaller, from -2^{-B_c} to $+2^{-B_c}$, for some number of bits $B_c > 0$. If the filters are implemented in custom or semi-custom hardware, this observation can be used to reduce the number of bits allocated to $\boldsymbol{\delta}\mathbf{w}_2(n)$, reducing the complexity of all multiplications related to the second filter, both for computing the output and for updating the filter coefficients. In summary, this idea amounts to making the filters share the most significant bits in all coefficients, and adapting them with the step size of the fast filter. During convergence, when the difference between the filters is expected to be more significant, disregarding the most significant bits of the difference filter has the effect of working just as a weight transfer mechanism from the fast to the slow filter.

In order to take full advantage of the reduced wordlength used to store $\boldsymbol{\delta}\mathbf{w}_2(n)$, the output of the second filter and the difference filter update should be computed as shown in Table VI. Algorithm's steps 3 and 6 can be computed with a reduced wordlength. As can be seen, all the costly operations related to the second filter are computed with a reduced wordlength. Using this technique, it is possible to use for the difference filter a wordlength of about half that of the first filter, with little to no degradation in performance, compared to using full wordlength for all variables.

VIII. SIGNAL MODALITY CHARACTERIZATION

As we have already discussed, one of the main advantages of the combination approach to adaptive filters is its versatility. In the rest of this tutorial we illustrate the applicability of this strategy in different fields where adaptive filters are generally used.

As a first application example, combination of adaptive filters can be employed to characterize signal modality, for instance, whether a particular signal is generated through a linear or non-linear process. This observation has been successfully applied to track changes in the modality of different kinds of signals,
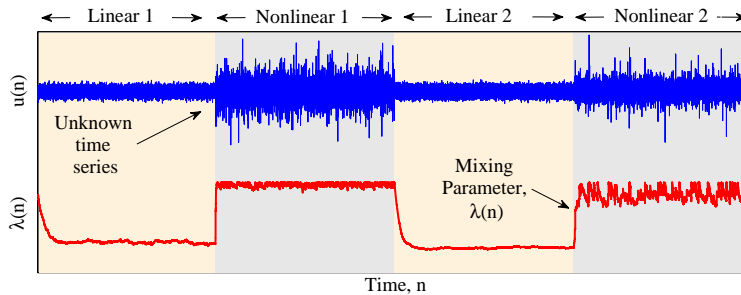
Fig. 10.   Signal modeling characterization using a combination of a linear and a non-linear filters. As can be seen, the mixing parameter is an effective indicator of the linear/non-linear nature of the input signal generation process.

including EEG [55], complex representations of wind and maritime radar [56], or speech [22]. To illustrate the idea, we consider the characterization of the linear/nonlinear nature of a signal along the lines of [22], which uses a convex combination of a normalized LMS (NLMS) filter and a normalized nonlinear gradient descent (NNGD) algorithm to predict future values of the input signal

$$\hat{u}(n+1) = \lambda(n)\hat{u}_{\text{NLMS}}(n+1) + [1 - \lambda(n)]\hat{u}_{\text{NNGD}}(n+1).$$

The mixing parameter of the combination, $\lambda(n)$, is adapted to minimize the square error of the prediction. A value $\lambda(n) \approx 1$ means that an NLMS filter suffices to achieve a good prediction allowing us to conclude that the input signal $u(n)$ is intrinsically linear. In the other extreme, when $\lambda(n)$ approaches 0, the predominance of the NNGD prediction suggests a non-linear input signal. Figure 10 shows the time evolution of the mixing parameter for a single realization of the algorithm, when we evaluate a signal that alternates between linear autoregressive processes (of orders 1 and 4) and two benchmark nonlinear signals described in [22, Eqs. (9)–(12)]. As it can be seen, information about the linear/nonlinear nature of the signal can be easily extracted from the evolution of the mixing parameter $\lambda(n)$.

IX. ADAPTIVE BLIND EQUALIZATION

Adaptive equalizers are widely used in digital communications to remove the intersymbol interference introduced by dispersive channels. In order to avoid the transmission of pilot sequences and use the channel bandwidth in an efficient manner, these equalizers can be initially adapted using a blind algorithm and switched to a decision-directed (DD) mode after the blind equalization achieves a sufficiently low steady-state mean-square error (MSE). The selection of an appropriate switching threshold is crucial, and this selection depends on many factors such as the signal constellation, the communication channel, or the SNR.

An adaptive combination of blind and DD equalization modes can be used, where the combination layer is itself adapted using a blind criterion. This combination scheme provides an automatic mechanism for
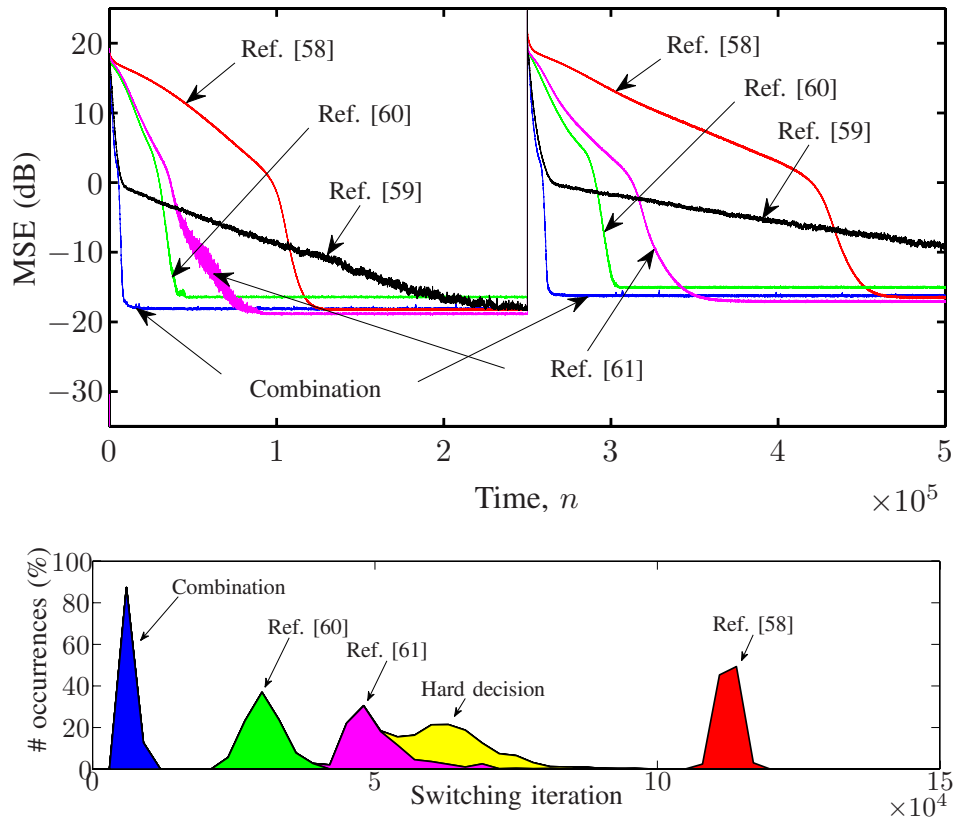
Fig. 11.   Peformance of blind equalization methods. Top panel: MSE time evolution averaged over 1000 runs. Bottom panel: Distribution of switching times from blind to DD mode (-16 dB).

smoothly switching between the blind and DD modes. The blind algorithm mitigates intersymbol interference during the initial convergence or when abrupt changes in the channel occur. Then, when the steady-state MSE is sufficiently low, the overall equalizer automatically switches to the DD mode to get an even lower error. The main advantage of this scheme is that it avoids the need to set *a priori* the transition MSE level [57], and automatically changes to DD mode as soon as a sufficient equalization level is attained.

Figure 11 illustrates the performance of the combination scheme as well as of other well-known schemes for blind equalization [58]–[61]. In all cases, the multimodulus algorithm (MMA) and the LMS algorithm were used for the blind and DD equalization phases, respectively. The communication scenario considers the transmission of a 256-QAM signal through a telephonic channel with an SNR of 40 dB, including an abrupt change at $n = 2.5 \times 10^5$. The top panel of Fig. 11 shows the MSE evolution for all methods averaged over 1000 independent runs, whereas the bottom plot illustrates the distribution of the switching time between blind and DD modes, considering that the switching to DD mode is complete when the MSE reaches -16 dB for a particular run. We can conclude that the combination scheme does a better job at detecting the

right moment for commuting to DD mode. Furthermore, the smaller variance observed in the switching iteration suggests that the combination method is more robust than other methods to the particularities of the simulation scenario.

This example shows that the combination approach can be used beyond MSE adaptive schemes. Here, we considered blind cost functions that exploit properties of high-order moments of the involved signals, both at the individual filter and at the combination layer levels. Similar uses can be conceived in other applications where high-order moments are frequently used, such as blind source separation.

X. SPARSE SYSTEM IDENTIFICATION

Sparse systems have gathered a lot of attention. These systems are characterized by long impulse responses with only a few nonzero coefficients, and are frequently encountered in applications such as network and acoustic echo cancellation [62], compressed sensing [63], High-Definition TV [64] and wireless communication and MIMO channel estimation [65], [66], among many others.

In the literature, several adaptive schemes have exploited such prior information to accelerate the convergence towards the sparse optimum solution. For instance, proportionate adaptive filters assign to each coefficient a different step size proportional to its amplitude [23], [67]. A recent approach [68] promotes sparsity based on adaptive projections where the sparsity constraints are included considering ℓ_1 balls and giving rise to a convex set whose shape is a hyperslab. An important family of sparse adaptive algorithms [69]–[71] incorporates sparsity enforcing norms into the cost function minimized by the filter.

The Zero-Attracting LMS (ZA-LMS) filter [69] uses a stochastic gradient descent rule in order to minimize a cost function that mixes the square power error and the ℓ_1-norm penalty of the weight vector. The update equation for this adaptive scheme reads:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{u}(n) - \rho \, \mathrm{sgn}[\mathbf{w}(n)], \tag{24}$$

where function sgn[·] extracts the sign of each element of the vector, and parameter ρ controls how the sparsity is favored in the coefficient adaptation. In fact, $\rho = 0$ turns (24) into the standard LMS update, removing any ability to promote sparsity. This scheme has shown improved performance with respect to the standard LMS algorithm when the unknown system is highly sparse; however, standard LMS outperforms ZA-LMS scheme when the system is dispersive. For this reason, there exists a tradeoff related to the degree of sparsity of the system, which unfortunately is usually unknown *a priori*, or can even vary over time.

In this subsection, we cover two approaches specially designed to alleviate this compromise:

- Scheme A: An adaptive convex combination of the ZA-NLMS filter, i.e., a sparsity-norm regularized version of the NLMS scheme and a standard NLMS algorithm [72]. This approach constitutes the
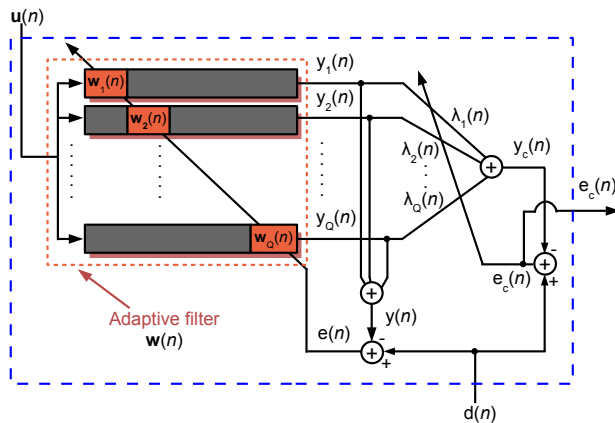
Fig. 12. Block diagram of Scheme B. Each block of the adaptive filter $\mathbf{w}(n)$ is represented as a complete filter computing its output using just the indicated coefficients.

straightforward application of combination schemes in order to alleviate the compromise regarding the selection of parameter ρ in the ZA-NLMS filter.

- Scheme B: A block-wise biased adaptive filter, depicted in Fig. 12, where the outputs of Q non-overlapping blocks of coefficients of an adaptive filter are weighted by different scaling factors to obtain the overall output [73]

$$y(n) = \sum_{q=1}^{Q} \lambda_q(n) y_q(n), \tag{25}$$

where $y_q(n)$ is the partial output of each block, and $\lambda_q(n) \in [0,1]$ with $q = 1, ..., Q$ are the shrinkage factors that adapt to minimize the power of $e(n) = d(n) - y(n)$. This scheme manages the well-known bias *vs* variance compromise in a block-wise manner [73]: if the mean-square deviation in the q-th block is much higher than the energy of the unknown system in this block, $\lambda_q(n)$ will tend to zero, biasing the output of the block $y_q(n)$ towards zero but reducing the output error. The shrinkage factors $\lambda_q(n)$ therefore act as estimators for the support (set of nonzero coefficients) of the filter.

Regarding the estimation of these scaling factors $\lambda_q(n)$, with $q = 1, ..., Q$, it can be shown that their optimal values lie in interval $(0, 1)$ [73]. Therefore, proceeding as in [38], we can reinterpret each element in the sum of (25) as a convex combination between $y_q(n)$ and a virtual filter whose output remains constant and equal to zero. This is useful, because it implies that we can employ rules similar to *cvx-LMS* and *cvx-PN-LMS* reviewed in Section IV for adjusting parameters $\lambda_q(n)$. It should be noted that, for this kind of scheme, the ability to model sparse systems increases with the number of blocks Q, since the length of each block decreases. However, the computational cost associated with
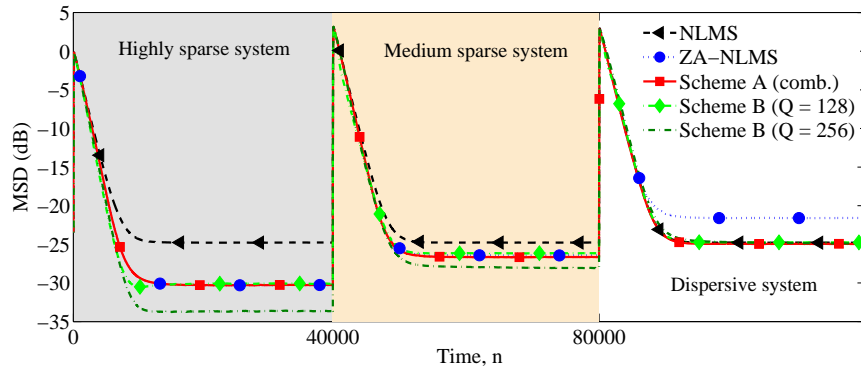
Fig. 13. MSD performance of both proposed algorithms: Scheme A (and its constituent filters: An NLMS filter with $\mu = 0.5$ and a ZA-NLMS with $\mu = 0.5$ and $\rho = 10^{-6}$) and Scheme B (based on an NLMS filter with $\mu = 0.5$) for two different configurations: $Q = 128$ and $Q = 256$ blocks. The sparseness of the plant varies over time.

the adaptation of the mixing parameters also increases with Q.

To illustrate both approaches, we have carried out a system identification experiment with an unknown plant with 1024 taps, whose sparsity degree changes over time considering white noise as input signal and SNR = 20 dB. We start with a very sparse system (only 16 active taps), that abruptly changes at $n = 40000$ to a plant with 128 active coefficients, and finally, at $n = 80000$, we employ a more dispersive unknown system with 512 active taps. Figure 13 shows the MSD reached by Scheme A and Scheme B, considering two different possibilities for the number of blocks Q.

As expected, Scheme A shows a robust behavior with respect to the sparsity degree of the unknown plant, behaving as well as its best component filter. Scheme B provides an attractive alternative, whose performance improves when the length of each block is reduced. Comparing both schemes, the computational cost of Scheme B is smaller than that of Scheme A, even when $Q = 256$. In this situation, we also observe that Scheme B outperforms Scheme A in terms of MSD for highly and medium sparse systems.

XI. ACOUSTIC ECHO CANCELLATION

Combination filters have been successfully employed in acoustic signal processing applications, such as active noise cancellation [30], [31], adaptive beamforming [29], and acoustic echo cancellation, where combination schemes have been used in linear cancellation (considering both time-domain [23] and frequency-domain [24] schemes), and in nonlinear acoustic echo cancellation [25], [26], [74].

The abundance of portable devices has been motivating the inclusion of nonlinear models in the adaptive scheme of acoustic echo cancellers (see Fig. 14), in order to compensate for the nonlinear distortion caused by low-cost loudspeakers fed by power amplifiers driven at high levels. One of the main solutions used for
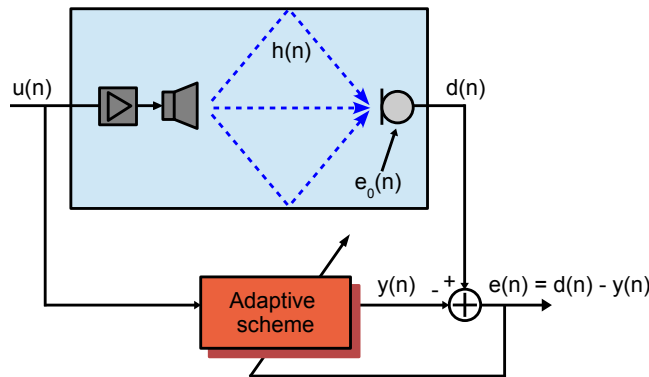
Fig. 14. Acoustic echo cancellation scenario. Signal $\mathbf{u}(n)$ stands for the input signal, $e_0(n)$ is the background noise at microphone location, $h(n)$ represents the room impulse response that describes the acoustic propagation (linear in nature), and where depending on the behavior of the loudspeaker and the power amplifier, nonlinear distortion can be found in desired signal $d(n)$. The adaptive scheme tries to minimize the power of the error cancellation $e(n)$, incorporating nonlinear models if necessary.

this purpose is the Volterra filter (VF), which is able to represent a large class of nonlinearities with memory [75].

The output of a VF can be calculated as the addition of partial outputs of different kernels, each one representing an order of nonlinearity, i.e.,

$$y(n) = \sum_{p=1}^{P} y_p(n) = \sum_{p=1}^{P} \sum_{i_1=0}^{N_p-1} \cdots \sum_{i_p=i_p-1}^{N_p-1} h_{p,i_1,\ldots,i_p} \prod_{q=1}^{p} u(n-i_q), \qquad (26)$$

where $y_p(n)$ corresponds with the output of the p-th kernel with length N_p, P is the number of kernels that compose the VF and $h_{p,i_1,\ldots,i_p}$ represents the adaptive coefficients of the p-th kernel. For instance, the output of a VF of order $P = 2$ can be obtained as $y(n) = y_1(n) + y_2(n)$, where $y_1(n)$ is the output of the linear kernel, and $y_2(n)$ represents the output of the quadratic kernel, responsible for the modeling of second-order nonlinearities. Despite the large computational cost associated with the operation of a VF, the success of this nonlinear filter lies in the simplicity of the adaptation of the coefficients in each kernel $h_{p,i_1,\ldots,i_p}$, which permits the employment of update rules similar to those of linear adaptive filters. For this reason, VFs present similar tradeoffs to those of linear filters, in particular the selection of the step size, and the memory of the kernels.

In [25], two different schemes were proposed to alleviate such compromises, named combination of Volterra filters and combination of kernels (CK) —see Fig. 15. The first approach constitutes a straightforward application of the combination idea, i.e., the output of the combined scheme is calculated mixing different VFs outputs, each one obtained by means of (26), and following any of the schemes included in Section VI. However, the second scheme proposes a special kind of VF, where each kernel is replaced by a combination

of kernels with complementary settings. For the case of a combination of two kernels per order, the output of the CK scheme reads

$$y(n) = \sum_{p=1}^{P} y_p(n) = \sum_{p=1}^{P} \lambda_p(n)y_{p,1}(n) + [1 - \lambda_p(n)]y_{p,2}(n), \qquad (27)$$

where the outputs of two kernels of order p, $y_{p,1}(n)$ and $y_{p,2}(n)$, are combined by means of the mixing parameter $\lambda_p(n)$. Both algorithms reach similar performance but CK is a more attractive scheme regarding the computational cost [25].

We should notice that the performance of VFs is also subject to a tradeoff related to the number of kernels, since the degree of nonlinearity present in the filtering scenario is rarely known *a priori*. This compromise directly affects the nonlinear acoustic echo cancellers: if only linear distortion is present in the echo path, the adaptation of nonlinear kernels degrades the performance of the whole VF, which behaves worse than a simple linear adaptive filter.

In order to deal with these compromises, a nonlinear acoustic echo canceller, represented in Figure 15 using both the combination of Volterra filters and the CK approaches, was proposed in [25] considering two requirements: the robustness with respect to the presence or absence of nonlinear distortion, and the fact that the acoustic room impulse response $h(n)$ is strongly variant. The first requirement is dealt with considering a combination of a quadratic kernel and a *virtual* kernel named *All-Zeros Kernel*, with all coefficients equal to zero and no adaptation, whose partial output is always zero. Regarding the second requisite, a combination of two linear kernels with different step sizes serves to provide a fast reconvergence and a low steady-state error.

The output of this algorithm, considering the CK scheme, is:

$$y(n) = \underbrace{\lambda_1(n)y_{1,1}(n) + [1 - \lambda_1(n)]y_{1,2}(n)}_{\text{linear part}} + \underbrace{\lambda_2(n)y_{2,1}(n) + [1 - \lambda_2(n)] \cdot 0}_{\text{nonlinear part}}, \qquad (28)$$

where $y_{1,1}(n)$ and $y_{1,2}(n)$ are the outputs of two linear kernels with different step sizes, $\lambda_1(n)$ is the mixing parameter to combine them, $y_{2,1}(n)$ is the output of the quadratic kernel, and $\lambda_2(n)$ is the mixing parameter to weight the estimation of the nonlinear echo.

Figure 16 represents the performance of the proposed nonlinear acoustic echo canceller, compared with that of all the adaptive filters (linear and nonlinear) that could be designed with its kernels, in terms of the Echo Return Loss Enhancement (ERLE) computed as:

$$\text{ERLE}(n) \triangleq 10\log \frac{\mathbb{E}\{[d(n) - e_0(n)]^2\}}{\mathbb{E}\{[e(n) - e_0(n)]^2\}}. \qquad (29)$$

As it can be seen, the proposed scheme reaches the best performance, behaving as a combination of linear filters when only linear distortion is present ($t \leq 30$ s). In this case, $\lambda_2(n) \approx 0$ and the quadratic kernel is
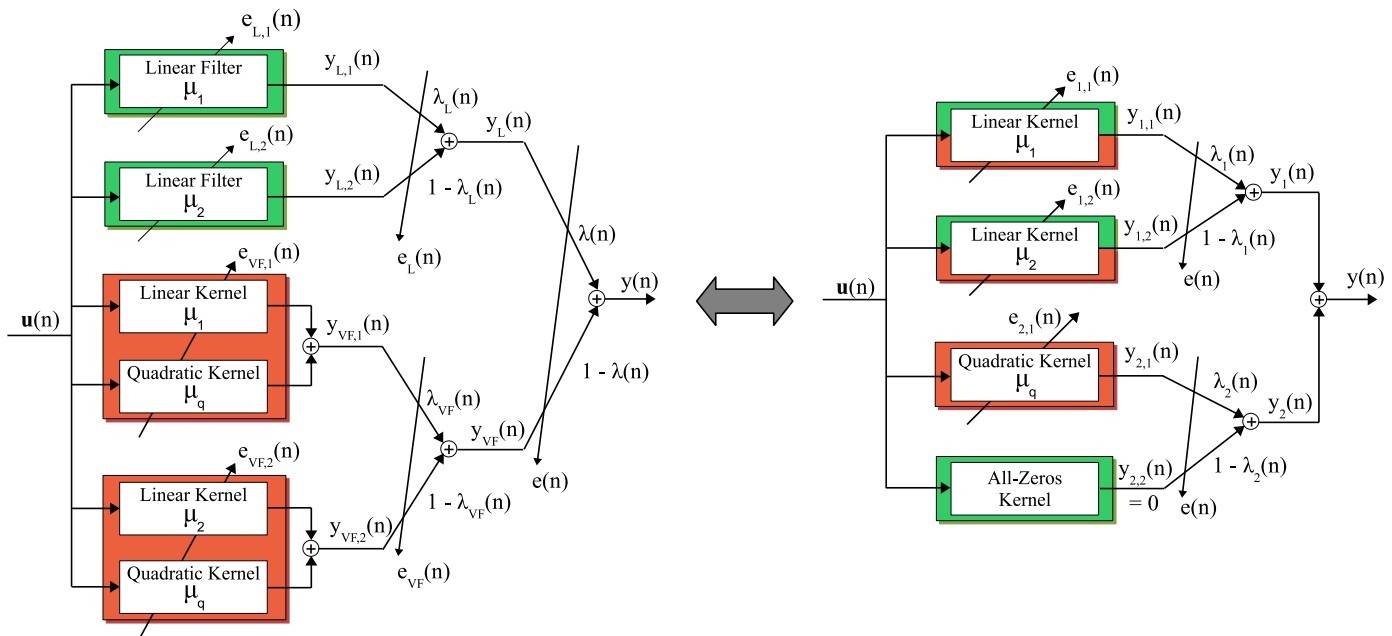
Fig. 15.   Nonlinear acoustic echo canceller based on combination of filters. On the left: A hierarchical combination of filters (linear and VFs) where each filter adapts employing its own error signal, for instance $e_{L,1}(n) = d(n) - y_{L,1}(n)$. On the right: A special VF based on the CK, where each kernel adapts using an error signal employing its own output and the combined outputs of the kernels of different orders (see [25]). The mixing parameters $\lambda_1(n)$ and $\lambda_2(n)$ update to minimize the power of the global error signal, i.e., $e(n) = d(n) - y(n)$.
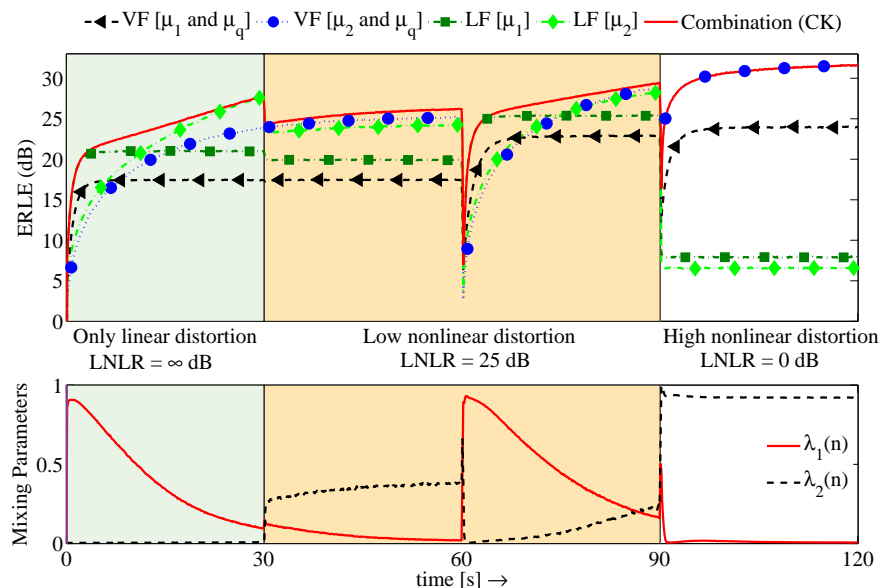

Fig. 16.   Performance of the proposed scheme as a function of the linear-to-nonlinear ratio of echo powers (LNLR), including an abrupt change in the room impulse response at $t = 60$ s. From top to bottom: ERLE achieved by the combination of kernels scheme and of all possible filters (linear and VFs) using its constituent kernels; time evolution of the mixing parameters. In this experiment, we employ Laplacian speech-like colored noise as input signal; experiments with real speech can be found in [25].

not considered in the output of the filter, without degrading the performance of the proposed scheme with respect to that of linear filters. At the end of the experiment ($t > 90$ s), when strong nonlinear distortion is considered, $\lambda_2(n) \approx 1$, and the output of the quadratic kernel is fully incorporated, providing the ability to model nonlinearities. In an intermediate case, during 30 s $< t \leq 90$ s, the proposed scheme slightly outperforms both linear and nonlinear filters. In addition, the algorithm shows a suitable reconvergence when the room impulse response abruptly changes, thanks to the combination of two linear kernels with different step sizes, as it can be seen after $t = 60$ s.

XII. CONCLUSIONS AND OPEN PROBLEMS

Combinations of adaptive filters constitute a powerful approach to improve the performance of adaptive filters. In this paper, we have reviewed some of the most popular combination schemes, highlighting their theoretical properties and limits. Practical algorithms to combine adaptive filters need to implement estimation methods to adjust the combination layer, taking into account the possibly time-varying conditions in which the filter operates.

We have reviewed several of the methods that have been proposed in the literature, paying special attention to gradient methods. Power-normalized algorithms are particularly interesting, since they simplify the selection of parameters and result in a more robust behavior when the statistics of the filtering scenario are (partly) unknown, which is frequently the case. The versatility of the approach has been demonstrated through several examples in a variety of applications. We have seen that, in all studied scenarios, combination schemes offer competitive performance when compared to state-of-the-art methods for each application. This fact, together with the inherent simplicity of the approach, make combination structures attractive for demanding applications requiring enhanced performance, as illustrated by the several examples and references mentioned in the body of the article.

In our opinion, some of the most interesting open problems to be addressed are:

- the selection and design of component filters with reduced cross-EMSE, with the goal to minimize the overall EMSE,

- providing and exploiting strategies to reduce the cost of combined schemes, trying to develop new structures whose complexity is close to that of an individual filter, and

- extensions to other application domains where different kinds of compromises and performance tradeoffs may be present.

ACKNOWLEDGMENTS

REFERENCES

[1] E. Hänsler and G. Schmidt (Eds.), *Topics in Acoustic Echo and Noise Control*. Berlin, Germany: Springer, 2006.

[2] B. Widrow, "A microphone array for hearing aids," *IEEE Circuits and Systems Mag.*, vol. 1, no.2, pp. 26–32, 2001.

[3] C. R. Johnson Jr., et al., "Blind equalization using the constant modulus criterion: A review," *Proc. IEEE*, vol. 86, no. 10, pp. 1927–1950, Oct. 1998.

[4] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[5] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, issue 4–5, pp. 311–801. Boston-Delft: NOW Publishers, 2014.

[6] A. H. Sayed, *Adaptive Filters*. NJ: Wiley, 2008.

[7] P. Andersson, "Adaptive forgetting factor in recursive identification through multiple models," *Intl. Journal of Control*, vol. 42, no. 5, pp. 1175–1193, May. 1985.

[8] M. Niedźwiecki, "Multiple-model approach to finite memory adaptive filtering," *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 470–473, Feb. 1990.

[9] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2685–2699, Oct. 1999.

[10] S. S. Kozat and A. C. Singer "Multi-stage adaptive signal processing algorithms," in *Proc. SAM Signal Process. Workshop*, Cambridge, MA, Mar. 16–17, 2000, pp. 380–384.

[11] M. Martínez-Ramón, J. Arenas-García, A. Navia-Vázquez, and A. R. Figueiras-Vidal, "An adaptive combination of adaptive filters for plant identification," in *Proc. 14th IEEE Intl. Conf. Digital Signal Process.*, Santorini, Greece, Jul. 2002, pp. 1195-1198.

[12] 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, no. 3, pp. 1078–1090, Mar. 2006.

[13] Y. Zhang and J. Chambers, "Convex combination of adaptive filters for a variable tap-length LMS algorithm," *IEEE Signal Process. Lett.*, vol. 13, no. 10, pp. 628–631, Oct. 2006.

[14] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Tracking properties of convex combinations of adaptive filters," *Proc. IEEE Workshop on Statistical Signal Processing (SSP)*, Bordeaux, France, Jul. 17–20, 2005, pp. 109-114.

[15] M. T. M. Silva and V. H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3137–3149, Jul. 2008.

[16] N. J. Bershad, J. C. M. Bermudez, and J.-Y. Tourneret, "An affine combination of two LMS adaptive filters – Transient mean-square analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1853–1864, May 2008.

[17] J. C. M. Bermudez, N. J. Bershad, and J.-Y. Tourneret, "Stochastic analysis of an error power ratio scheme applied to the affine combination of two LMS adaptive filters," *Signal Process.*, vol. 91, no. 11, pp. 2615–2622, Nov. 2011.

[18] R. Candido, M. T. M. Silva, and V. H. Nascimento, "Transient and steady-state analysis of the affine combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 4064–4078, Jul. 2010.

[19] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, "Steady-state MSE performance analysis of mixture approaches to adaptive filtering," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4050–4063, Aug. 2010.

[20] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, "Transient analysis of adaptive affine combinations," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6227–6232, Dec. 2011.

[21] T. Trump, "Transient analysis of an output signal based combination of two NLMS adaptive filters," in *Proc. Intl. Workshop on Mach. Learning for Signal Process.*, Kittilä, Finland, Aug. 29–Sep. 01, 2010, pp. 244–249.

[22] B. Jelfs, S. Javidi, P. Vayanos, and D. P. Mandic, "Characterisation of signal modality: Exploiting signal nonlinearity in machine learning and signal processing," *Journal of Signal Process. Systems*, vol. 61, no. 1, pp. 105–115, Oct. 2010.

[23] J. Arenas-García, and A. R. Figueiras-Vidal, "Adaptive combination of proportionate filters for sparse echo cancellation," *IEEE Trans. Audio, Speech and Language Process.*, vol. 17, no. 6, pp. 1087–1098, Aug. 2009.

[24] J. Ni and F. Li, "Adaptive combination of subband adaptive filters for acoustic echo cancellation," *IEEE Trans. Consumer Electronics*, vol. 56, no. 3, pp. 1549–1555, Aug. 2010.

[25] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-García, and W. Kellermann, "Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation," *IEEE Trans. Audio, Speech and Language Process.*, vol. 19, no. 1, pp. 97–110, Jan. 2011.

[26] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, W. Kellermann, and J. Arenas-García, "Enhanced adaptive Volterra filtering by automatic attenuation of memory regions and its application to acoustic echo cancellation," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2745–2750, Jun. 2013.

[27] T. Trump, "A combination of two NLMS filters in an adaptive line enhancer," *Proc. Intl. Conf. Digital Signal Process.*, Corfu, Greece, Jul. 6–8, 2011, pp. 1–6.

[28] S. Lu, J. Sun, G. Wang, and J. Tian, "A novel GSC beamformer using a combination of two adaptive filters for smart antenna array," *IEEE Antennas and Wireless Propagation Lett.*, vol. 11, pp. 377–380, Apr. 2012.

[29] D. Comminiello, M. Scarpiniti, R. Parisi, and A. Uncini "Combined adaptive beamforming schemes for nonstationary interfering noise reduction," *Signal Processing*, vol. 93, no. 12, pp. 3306–3318, Dec. 2013.

[30] M. Ferrer, A. Gonzalez, M. de Diego, and G. Piñero, "Convex combination filtered-X algorithms for active noise control systems," *IEEE Trans. Audio, Speech and Language Process.*, vol. 21, no. 1, pp. 156–167, Jan. 2013.

[31] N. V. George, and A. Gonzalez, "Convex combination of nonlinear adaptive filters for active noise control," *Applied Acoustics.*, vol. 76, pp. 157–161, Feb. 2014.

[32] L. Kuncheva, *Combining Pattern Classifiers*. NJ: Wiley, 2004.

[33] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, 1997.

[34] V. Vovk, "Competitive on-line statistics," *Int. Statist. Rev.*, vol. 69, no. 2, pp. 213–248, 2001.

[35] H. Ozkan, M. A. Donmez, S. Tunc and S. S. Kozat, "A deterministic analysis of an online convex mixture of experts algorithm," *IEEE Trans. Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1575–1580, Jul. 2015.

[36] V. H. Nascimento, M. T. M. Silva, L. A. Azpicueta-Ruiz, and J. Arenas-García, "On the tracking performance of combinations of least mean squares and recursive least squares adaptive filters," in *Proc. ICASSP*, Dallas, TX, Mar. 14–19, 2010, pp. 3710–3713.

[37] L. A. Azpicueta-Ruiz, A.R. Figueiras-Vidal, and J. Arenas-García, "A normalized adaptation scheme for the convex combination of two adaptive filters," in *Proc. ICASSP*, Dallas, USA, Mar. 31–Apr. 04, 2008, pp. 3301–3304.

[38] M. Lázaro-Gredilla, L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-García, "Adaptively biasing the weights of adaptive filters," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3890–3895, Jul. 2010.

[39] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-García, "A new least squares adaptation scheme for the affine combination of two adaptive filters," in *Proc. Intl. Workshop on Mach. Learning for Signal Process.*, Cancun, Mexico, Oct. 16–19, 2008, pp. 327–332.

[40] T. Trump, "Output statistics of a line enhancer based on a combination of two adaptive filters," *Central European Journal of Engineering*, vol. 1, no. 3, pp. 244–252, Sep. 2011.

[41] L. Shi, Y. Lin, and X. Xie, "Combination of affine projection sign algorithms for robust adaptive filtering in non-Gaussian impulsive interference," *Electronics Lett.*, vol. 50, no. 6, pp. 466–467, Mar. 2014.

[42] R. Harris, D. Chabries, and F. Bishop, "A variable step (VS) adaptive filter algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 2, pp. 309–316, Apr. 1986.

[43] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 7, pp. 1633–1642, Jul. 1992.

[44] V. J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Signal Process.*, vol. 41, no. 6, pp. 2075–2087, Jun. 1993.

[45] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-type algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 631–639, Mar. 1997.

[46] M. H. Costa and J. C. M. Bermudez, "A robust variable step size algorithm for LMS adaptive filters," in *Proc. ICASSP*, Toulouse, France, May 14–19, 2006, vol. III, pp. 93–96.

[47] 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, no. 9, pp. 2430–2438, Sep. 2006.

[48] V. H. Nascimento and R. C. de Lamare, "A low-complexity strategy for speeding up the convergence of convex combinations of adaptive filters," in *Proc. ICASSP*, Kyoto, Japan, Mar. 25–30, 2012, pp. 3553–3556.

[49] L. F. O. Chamon, W. B. Lopes, and C. G. Lopes, "Combination of adaptive filters with coefficients feedback," in *Proc. IEEE ICASSP*, Kyoto, Japan, Mar. 25–30, 2012, pp. 3785–3788.

[50] L. A. Azpicueta-Ruiz, M. Zeller, A.R. Figueiras-Vidal, and J. Arenas-García, "Least-squares adaptation of affine combinations of multiple adaptive filters," in *Proc. ISCAS*, Paris, France, 2010, pp. 2976–2979.

[51] J. Arenas-García, V. Gómez-Verdejo, and A. R. Figueiras-Vidal, "New algorithms for improved adaptive convex combination of LMS transversal filters," *IEEE Trans. Instrumentation and Measurement.*, vol. 54, no. 6, pp. 2239–2249, Dec. 2005.

[52] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford University Press, 1995.

[53] L. A. Azpicueta-Ruiz, J. Arenas-Garcia, V. H. Nascimento, and M. T. M. Silva, "Reduced-cost combination of adaptive filters for acoustic echo cancellation," in *Proc. Intl. Telecommunications Symposium (ITS)*, Sao Paulo, Brazil, Aug. 17–20, 2014, pp. 1–5.

[54] V. H. Nascimento, M. T. M. Silva, and J. Arenas-García, "A low-cost implementation strategy for combinations of adaptive filters," in *Proc. IEEE ICASSP*, Vancouver, Canada, May 26–31, 2013, pp. 5671–5675.

[55] L. Li, Y. Xia, B. Jelfs, and D. P. Mandic, "Modelling of brain consciousness based on collaborative filters," *Neurocomputing*, vol. 76, no. 1, pp. 36–43, Jan. 2012.

[56] B. Jelfs, D. P. Mandic, and S. C. Douglas "An adaptive approach for the identification of improper signals," *Signal Processing*, vol. 92, no. 2, pp. 335–344, Feb. 2012.

[57] M. T. M. Silva and J. Arenas-García, "A soft-switching blind equalization scheme via convex combination of adaptive filters," *IEEE Trans. Signal Process.*, vol. 61, no. 5, pp. 1171–1182, Mar. 2013.

[58] G. Picchi and G Prati, "Blind equalization and carrier recovery using a "stop-and-go" decision-directed algorithm," *IEEE Trans. Commun.*, vol. 35, no. 9, pp. 877–887, Sep. 1987.

[59] V. Weerackody and S.A. Kassam, "Dual-mode type algorithms for blind equalization," *IEEE Trans. Commun.*, vol. 42, no. 1, pp. 22–28, Jan. 1994.

[60] F. C. C. De Castro, M. C. F. De Castro, and D. S. Arantes, "Concurrent blind deconvolution for channel equalization," in *Proc. of ICC*, vol. 2, Helsinki, Finland, Jun. 11–14, 2001, pp. 366–371.

[61] J. Mendes-Filho, M. D. Miranda, and M. T. M. Silva, "A regional multimodulus algorithm for blind equalization of QAM signals: Introduction and steady-state analysis," *Signal Processing*, vol. 92, no. 11, pp. 2643–2656, Nov. 2012.

[62] C. Paleologu, J. Benesty and S. Ciochina, *Sparse Adaptive Filters for Echo Cancellation*, Morgan & Claypool, 2011.

[63] N. Kalouptsidis, G. Mileounis, B. Babadi and V. Tarokh, "Adaptive algorithms for sparse system identification," *Signal Process.*, vol. 91, no. 8, pp. 1910–1919, Aug. 2011.

[64] W. Schreiber, "Advanced television systems for terrestrial broadcasting: Some problems and some proposed solutions," *Proceedings of the IEEE.*, vol. 83, no. 6, pp. 958–981, Jun. 1995.

[65] Guan Gui and Li Xu, "Affine combination of two adaptive sparse filters for estimating large scale MIMO channels," in *Proc. Annual Summit and Conference Asia-Pacific Signal and Information Processing Association (APSIPA)*, Chiang Mai, Thailand, Dec. 9–12, 2014 , pp. 1–7.

[66] Guan Gui, S. Kumagai, A. Mehbodniya, F. Adachi, and Li Xu, "Variable is good: Adaptive sparse channel estimation using VSS-ZA-NLMS algorithm," in *Proc. Intl. Conf. Wireless Communications & Signal Processing (WCSP)*, Hangzhou, China, Oct. 24–26, 2013, pp. 1–5.

[67] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Audio, Speech and Language Process.*, vol. 8, no. 5, pp. 508–518, Sep. 2000.

[68] Y. Kopsinis, K. Slavakis and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1 balls," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.

[69] Y. Chen, Y. Gu and A.O. Hero III, "Sparse LMS for system identification," in *Proc. ICASSP*, Taipei, Taiwan, Apr. 19–24, 2009, pp. 3125–3128.

[70] S. Zhang and J. Zhang, "Transient analysis of zero-attracting NLMS algorithm without Gaussian inputs assumption," *Signal Process.*, vol. 97, pp. 100–109, Apr. 2014.

[71] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[72] B. K. Das, and M. Chakraborty, "Sparse adaptive filtering by an adaptive convex combination of the LMS and the ZA-LMS algorithms," *IEEE Trans. Circuits and Systems - I: Regular papers*, vol. 61, no. 5, pp. 1499–1507, May 2014.

[73] L. A. Azpicueta-Ruiz, M. Lázaro-Gredilla, A.R. Figueiras-Vidal, and J. Arenas-García, "A block-based approach to adaptively bias the weights of adaptive filters," in *Proc. Intl. Workshop on Mach. Learning for Signal Process.*, Beijing, China, Sep. 18–21, 2011, pp. 1–6.

[74] D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-García, and A. Uncini, "Functional link adaptive filters for nonlinear acoustic echo cancellation," *IEEE Trans. Audio, Speech and Language Process.*, vol. 21, no. 7, pp. 1502–1511, Jul. 2013.

[75] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Process. Mag.*, vol. 8, no. 3, pp. 10–26, Jul. 1991.