uc3m | Universidad Carlos III de Madrid

e-Archivo

This is a postprint version of the following published document:

Comminiello, D., Scarpiniti, M., Azpicueta-Ruiz, L. A., Arenas-García, J. & Uncini, A. (2017). Combined nonlinear filtering architectures involving sparse functional link adaptive filters. *Signal Processing, 135*, 168–178.

DOI: 10.1016/j.sigpro.2017.01.009

© 2017 Elsevier B.V. All rights reserved.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

Combined Nonlinear Filtering Architectures Involving Sparse Functional Link Adaptive Filters

Danilo Comminiello[a,*], Michele Scarpiniti[a], Luis A. Azpicueta-Ruiz[b], Jerónimo Arenas-García[b], and Aurelio Uncini[a]

[a] *Department of Information Engineering, Electronics and Telecommunications (DIET), "Sapienza" University of Rome, 00184 Rome, Italy*
[b] *Department of Signal Theory and Communications, Universidad Carlos III de Madrid, 28911 Leganés, Spain*

Abstract

Sparsity phenomena in learning processes have been extensively studied, since their detection allows to derive suited regularized optimization algorithms capable of improving the overall learning performance. In this paper, we investigate the sparsity behavior that may occur in nonlinear adaptive filtering problems and how to leverage it and develop enhanced algorithms. In particular, we focus on a particular class of linear-in-the-parameters nonlinear adaptive filters, whose nonlinear transformation is based on a functional link expansion. The analysis of the sparsity in functional links leads us to derive a family of adaptive combined filtering architectures that is capable of exploiting any sparseness degree in the nonlinear filtering. We propose two different filtering schemes based on a new block-based combined approach, well suited for sparse adaptive algorithms. Moreover, a hierarchical architecture is also proposed that generalizes the different combined schemes and does not need any *a priori* information about the nature of the nonlinearity to be modeled. Experimental results prove the effectiveness of the proposed combined architectures in exploiting any sparseness degree and improving the modeling performance in nonlinear system identification problems.

Keywords: Nonlinear Adaptive Filtering, Functional Links, Linear-in-the-Parameters Nonlinear Filters, Sparse Adaptive Filters, Combination of Filters

*Corresponding author. Tel.: +39 06 44585495, Fax: +39 06 4873300.
E-mail address: danilo.comminiello@uniroma1.it.

December 19, 2016

1. Introduction

Sparsity is a widespread concept in several signal processing fields and it may concern and characterize a signal or even a system. In both cases, the awareness of any sparseness may be even very important, since it often leads to performance improvement or computational saving. A signal is characterized by a sparse representation when the whole signal is mostly described by only a few elementary portions of it. This property is exploited in several applications, ranging from pattern recognition to image and audio processing, to identify and characterize new observations of the signal by the knowledge of a few *atoms* of it [1–4]. On the other hand, sparsity may also characterize the impulse response of a system, and this occurs when a small percentage of its components has a significant magnitude while the rest is zero or negligible, i.e., very close to zero. From an energy point of view, it can be said that an impulse response is sparse if a large fraction of its energy is concentrated in a small fraction of its duration [5]. This property has been often exploited in several applications by regularizing the learning algorithm to improve the modeling performance, e.g., in wireless communications [6, 7], network and acoustic echo cancellation [8–10, 5], room acoustics [11], adaptive beamforming [12, 13], and underwater communications [14, 15], among others.

Sparsity has also been exploited in online learning for nonlinear signal processing. Nonlinear transformations of a signal may often introduce redundant or useless parameters, thus incurring in overfitting phenomena. Such a problem can be tackled by selecting the smallest subset of nonlinear transformations that provides the highest contribution in terms of nonlinear modeling performance [16]. In the recent literature, most representative examples of sparsity-aware nonlinear models include sparse algorithms for kernel methods [17–21], and for Volterra and polynomial filters [22, 23].

Recently, sparsity has been analyzed in a class of linear-in-the-parameters nonlinear filters, known as functional link adaptive filters (FLAFs) [24–26], which are characterized by a nonlinear expansion of the input followed by a linear filtering of the transformed signal. In FLAFs, the nonlinear transformation is performed by using nonlinear series expansions (e.g., trigonometric, Chebyshev or Legendre polynomial series [27], among others), satisfying universal approximation constraints [28]. Depending on the specific nonlinear

nature of the scenario that distorts an input signal, the functional links involved in an FLAF may show certain sparseness degree, i.e., only a selection of them is really useful to model a nonlinear system. This is the reason why the proportionate FLAFs have been introduced [26] to exploit such sparse functional links.

However, the sparseness of a functional link expansion may vary due to different factors, such as the level of nonlinearity affecting a signal or the filter design. Therefore, very often, it is difficult to provide a suitable configuration for a proportionate FLAF without any prior knowledge of the nonlinearity to be modeled. For this reason, in order to take always advantage of sparse functional links regardless of their sparseness degree, in this paper we propose the use of combined nonlinear filtering architectures involving different sparseness degree of functional links. Adaptive combination of filters is an effective technique to adequately select adaptive algorithm parameters, and it has been successfully exploited in different adaptive signal processing applications [29]. This approach permits to improve modeling performance when the *a priori* knowledge about a filtering scenario is unknown or limited [30–34, 29],[35], thus enhancing the robustness of adaptive schemes even in time-varying scenarios. The idea of using combined schemes to exploit the sparse nature of functional links was preliminary introduced in [36], where a combined FLAF scheme was presented involving a convex combination of two proportionate adaptive filters downstream of the functional expansion.

In this paper, we extend and improve such an idea, providing a deeper analysis of the sparse functional links and a new block-based approach for the combination of proportionate FLAFs. The choice of adopting a block-based combination approach derives from the analysis of the sparse functional links and, in particular, from the fact that coefficients with highest energy are adjacent to each other, so it is possible to select blocks with different sparseness degrees. In similar situations, block-based strategies have been shown to improve the performance [37, 32, 38, 29]. Here, we propose a new block-based strategy that takes into account the cyclic nature of sparse functional links. Based on this approach, we introduce two different combined schemes: the first one is characterized by a block-based convex combination of filters downstream of a shared functional expansion, while the second architecture involves the combination of two proportionate FLAFs each one including its own functional expansion. We also generalize the two proposed schemes by introducing a hierarchical combined architecture involving two stages of combinations of FLAFs. The proposed architectures are assessed

in different nonlinear system identification problems, in which we show the advantages of using one of the proposed architectures with respect to the other ones. Moreover, we prove the effectiveness of the hierarchical combined architecture, highlighting its generalization properties.

The rest of the paper is organized as follows: an analysis of the sparse nature of functional links is presented in Section 2, where it is shown how to exploit such sparseness to improve the modeling performance. Based on that analysis, in Section 3 we introduce a new block-based combination approach for sparse FLAFs, which leads to the derivation, in Section 4, of two different combined nonlinear filtering architectures and a hierarchical scheme that generalizes the previous ones. In Section 5, experimental results prove the effectiveness of the proposed models in different nonlinear modeling scenarios. Finally, in Section 6 our conclusions are drawn.

Notation

In this paper, matrices are represented by boldface capital letters and vectors are denoted by boldface lowercase letters. Time-varying vectors and matrices show discrete-time index as a subscript index, while in time-varying scalar elements the time index is denoted in square brackets. A regression vector is represented as $\mathbf{x}_n \in \mathbb{R}^M = \left[\begin{array}{cccc} x[n] & x[n-1] & \dots & x[n-M+1] \end{array} \right]^\top$, where M is the overall vector length and $x[n-i]$ is individual entry at the generic time instant $n-i$. A generic coefficient vector, in which all the elements depend on the same time instant, is denoted as $\mathbf{w}_n \in \mathbb{R}^M = \left[\begin{array}{cccc} w_0[n] & w_1[n] & \dots & w_{M-1}[n] \end{array} \right]^\top$, where $w_i[n]$ is the generic i-th individual entry at the n-th time instant. All vectors are represented as column vectors. The index related to a generic j-th filter is denoted as subscript, before the time index for vectors and matrices, e.g., $\mathbf{w}_{j,n}$. When dealing with blockwise filtering, we denote the l-th partition of the j-th filter as $\mathbf{w}_{j,n}^{(l)}$.

2. Exploiting Sparsity in Nonlinear Modeling by Functional Links

2.1. The FLAF Model

The functional link adaptive filter (FLAF), as presented in [24], is a linear-in-the-parameters nonlinear filter composed of a nonlinear functional expansion block (FEB) and a subsequent linear adaptive filter, as depicted in Fig. 1. The FEB consists of a series of functions satisfying universal approximation constraints, which are called *functional links*. The chosen set of functional

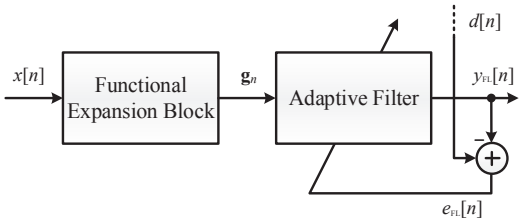


Figure 1: The nonlinear functional link adaptive filter.

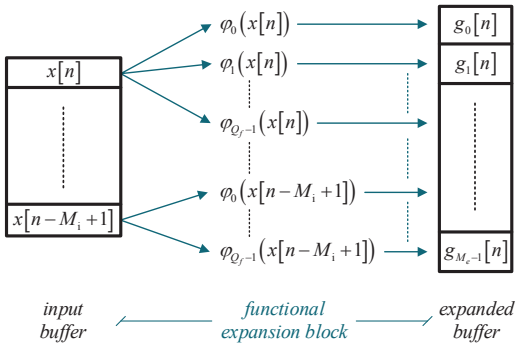


Figure 2: Memoryless functional link expansion.

links is denoted as $\Phi = \left\{\varphi_0(\cdot), \ldots, \varphi_{Q_{\mathrm{f}}-1}(\cdot)\right\}$, where Q_{f} is the number of functional links.

As depicted in Fig. 2, at the n-th time instant, the FEB receives an input frame, which can be a regression buffer $\mathbf{x}_n \in \mathbb{R}^{M_{\mathrm{i}}} = \begin{bmatrix} x[n] & x[n-1] & \ldots & x[n-M_{\mathrm{i}}+1] \end{bmatrix}^\top$, where M_{i} is defined as the input buffer length. Each sample of $\mathbf{x}_n$ is expanded by the chosen set of functions Φ, thus yielding the nonlinear expanded buffer $\mathbf{g}_n \in \mathbb{R}^{M_{\mathrm{e}}} = \begin{bmatrix} g_0[n] & g_1[n] & \ldots & g_{M_{\mathrm{e}}-1}[n] \end{bmatrix}^\top$, where $M_{\mathrm{e}} \geq M_{\mathrm{i}}$ represents the length of this expanded buffer.

Several choices can be made to build the functional link set (e.g., see [27]). A convenient and suitable choice for this study can be represented by the trigonometric series expansion, whose elements can be described by the following equations:

$$\varphi_j(x[n-i]) = \begin{cases} \sin(p\pi x[n-i]), & j = 2p-2 \\ \cos(p\pi x[n-i]), & j = 2p-1 \end{cases} \tag{1}$$

where $p = 1, \ldots, P$ is the expansion index, being P the expansion order, and $j = 0, \ldots, Q_{\mathrm{f}} - 1$ the functional link index. It is easy to verify that the

resulting set Φ is composed of $Q_{\mathrm{f}} = 2P$ functional links. It follows that the expanded buffer length can be derived as $M_{\mathrm{e}} = Q_{\mathrm{f}} M_{\mathrm{i}} = 2PM_{\mathrm{i}}$.

The expanded buffer $\mathbf{g}_n$ is composed of nonlinear elements only, since we want to use the FLAF only to estimate the nonlinear part of a signal or system impulse response. Note also that (1) actually refers to a *memoryless* expansion, since it does not involve cross-products of the n-th input sample with previous samples. However, the FEB can be easily extended to the case of functional links with memory (see [24] for a detailed explanation).

As depicted in Fig 1, the achieved expanded buffer $\mathbf{g}_n$ is then processed by an adaptive filter $\mathbf{w}_{\mathrm{FL},n} \in \mathbb{R}^{M_{\mathrm{e}}} = \left[\begin{array}{ccc} w_{\mathrm{FL},0}[n] & \dots & w_{\mathrm{FL},M_{\mathrm{e}}-1}[n] \end{array} \right]^{\top}$, thus providing the nonlinear FLAF output $y_{\mathrm{FL}}[n] = \mathbf{g}_n^{\top} \mathbf{w}_{\mathrm{FL},n-1}$. At each time instant, the coefficient vector $\mathbf{w}_{\mathrm{FL},n}$ can be updated by using any adaptive filtering algorithm (see for example [39, 40]).

It is worth noting that the main advantage of using trigonometric polynomial expansion is that in this way the functional links can be viewed as the partial sum of its Fourier series [41]. In fact, it is well known that among all the P-th order polynomials, the best approximation in the metric space $\mathcal{L}^2$ is given by the P-th partial sum of its Fourier series with respect to this system. Therefore, the functional links in (1) can be seen as a representation of the Fourier basis functions, while the related weight vector $\mathbf{w}_{\mathrm{FL},n}$ contains the coefficients of the Fourier series expansion.

2.2. Sparsity in Functional Links

In order to understand how sparsity appears in functional link transformations, we focus on the recursive equation of the trigonometric expansion (1) and on the coefficient vector $\mathbf{w}_{\mathrm{FL},n}$ that filters the expanded buffer. We want to know what are the functional links that mainly contribute to the estimation of a nonlinearity. To this end, we analyze the energy of the coefficient vector $\mathbf{w}_{\mathrm{FL},n}$ at steady state, i.e., for $n \to \infty$.

Consider one input sample to the FEB, i.e., $M_{\mathrm{i}} = 1$. The early coefficients of $\mathbf{w}_{\mathrm{FL},n}$ are those related to the functional links with small values of the expansion order p, while the last ones are related to the functional links with the highest values of p. If we look at the squared absolute value of $\mathbf{w}_{\mathrm{FL},n}$ we will usually notice a decreasing exponential behavior. The justification of this behavior can be easily derived by considering that, as previously said, $\mathbf{w}_{\mathrm{FL},n}$ represents the vector containing the coefficients of the Fourier series expansion. Indeed, let us consider a nonlinear distortion function of class C^k, continuous and differentiable on a closed interval. Then, it is well known [42]

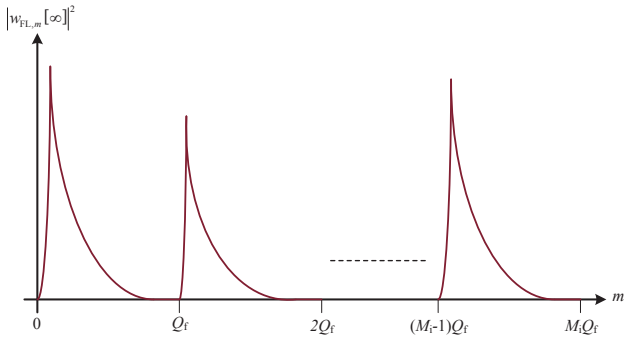


Figure 3: Sparse behavior of functional links.

that, in order to obtain the absolute and uniform convergence of the series, the coefficients must satisfy the following condition:

$$|w_{\mathrm{FL},m}[n]| \leq \frac{c}{m^{k+\theta}} \tag{2}$$

for any $c > 0$ and $\theta > 1$, and $m = 1, \ldots, M_\mathrm{e}$.

As a remark, this means that early functional links, i.e., those with small values of p, generate the most significant samples of the expanded frame, while the last functional links of a set Φ only produce small variations in a nonlinear modeling process, since high-order coefficients aim at modeling those nonlinear components that are not always present in a distortion (e.g., high-order harmonics), thus providing just an improvement of the modeling performance. It is worth noting that this energy decay is very similar to that of an acoustic impulse response, which is characterized by some early reflections that bring the largest fraction of its energy [5, 32].

Sparseness degree in functional links expansions may vary according to the nonlinearity to be estimated and to the expansion order P. If we use a small expansion order (e.g., $P < 3$) to estimate a strong nonlinearity, all the functional links involved will prove to be significant for the purpose of modeling. In this case, the functional link expansion will show a low degree of sparseness, i.e., the energy of the coefficient vector will show significant values, far from zero. However, if we use a larger expansion order ($P > 3$) to increase the approximation capabilities, very often we will incur in a sparse functional link expansion when nonlinearities are mild. This means that the energy of the coefficient vector will clearly show its exponential decay. In short, for strong nonlinearity we may need a large value of P (even larger

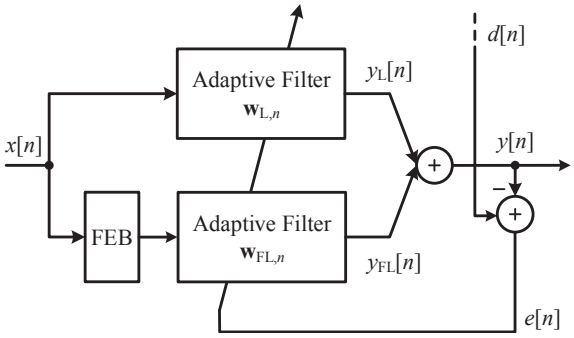


Figure 4: The split functional link adaptive filter.

than 10), while for mild nonlinearities an expansion order $P > 3$ is sufficient to generate a sparse functional link behavior. Usually, if we use an expansion order $P = 10$, which is a recommended value for a lot of applications, we will notice some sparsity in the functional link expansion. Overall, the larger the expansion order, the longer the tail of the energy decay in $\mathbf{w}_{\mathrm{FL},n}$.

If we generalize the above analysis to the case of interest, i.e., when the input to the FEB is a buffer with $M_{\mathrm{i}} > 1$, the energy of the coefficient vector $\mathbf{w}_{\mathrm{FL},n}$ is characterized by a periodic exponential decay, where the number of repetitions will coincide with the number of input samples to the FEB, i.e., M_{i}. For each input sample, we may have a different energy absolute value, but a similar behavior regarding the energy decay, as illustrated in Fig. 3. The relationship between the nonlinearity of a system and the expansion order with the sparseness degree of the corresponding functional link expansions will be also highlighted and supported by experimental results in Section 5.

2.3. Proportionate Split Functional Link Adaptive Filters

Very often in real-world problems, the response of a system to be identified is produced by any combination of a linear and nonlinear components. In order to model such kind of systems, the split functional link adaptive filter (SFLAF) architecture was recently proposed [24]. The SFLAF, depicted in Fig. 4, is a parallel filtering structure including a linear branch and a nonlinear branch. The former is simply composed of a linear adaptive filter, which is devoted to modeling the linear components of an unknown system. This allows the nonlinear branch, which is the nonlinear model of Fig. 1, to be focused on the modeling of the nonlinear components of the system.

An input sample $x\,[n]$ to the SFLAF is stored in the input buffer $\mathbf{x}_n \in$

$\mathbb{R}^M = \left[\; x\left[n\right] \;\; \ldots \;\; x\left[n-M+1\right] \; \right]^\top$, where M is the length of the adaptive filter $\mathbf{w}_{\mathrm{L},n} = \left[\; w_{\mathrm{L},0}\left[n\right] \;\; \ldots \;\; w_{\mathrm{L},M-1}\left[n\right] \; \right]^\top$. The adaptive filtering on the linear branch yields the output $y_{\mathrm{L}}\left[n\right] = \mathbf{x}_n^\top \mathbf{w}_{\mathrm{L},n-1}$, which has linear nature. The first M_{i} samples of $\mathbf{x}_n$ are also selected to be processed by the FEB, as described in Subsection 2.1.

The overall error signal of the SFLAF is:

$$\begin{aligned} e\left[n\right] &= d\left[n\right] - y\left[n\right] \\ &= d\left[n\right] - \mathbf{x}_n^\top \mathbf{w}_{\mathrm{L},n-1} - \mathbf{g}_n^\top \mathbf{w}_{\mathrm{FL},n-1} \end{aligned} \tag{3}$$

where $y\left[n\right]$ is the overall SFLAF output signal, resulting from the sum of the two branch outputs. Both adaptive filters try to minimize the power of $e\left[n\right]$, and this can be realized following different adaptation schemes. However, in this paper, in order to focus on the nonlinear branch we choose a classic normalized least-mean square (NLMS) algorithm for $\mathbf{w}_{\mathrm{L},n}$ (see also [26]).

When nonlinearities introduced by the unknown system are varying in time, the nonlinear elements generated by the FEB and stored in $\mathbf{g}_n$ may be not all useful in the same way for the modeling, and this may cause a performance decrease. A possible solution is the use of a regularizing weighted mask for the filter of the nonlinear branch [26], in an attempt to give more prominence to those nonlinear elements of the expanded buffer that have an active role in the modeling of nonlinearities. This means taking advantage of functional link sparsity. To this end, the update equation of $\mathbf{w}_{\mathrm{FL},n}$ can be written as [26]:

$$\mathbf{w}_{\mathrm{FL},n} = \mathbf{w}_{\mathrm{FL},n-1} + \mu_{\mathrm{FL}} \frac{\mathbf{Q}_n \mathbf{g}_n}{\mathbf{g}_n^\top \mathbf{Q}_n \mathbf{g}_n + \delta_{\mathrm{PFL}}} e\left[n\right] \tag{4}$$

which is the solution of the the following optimization problem:

$$\begin{aligned} &\arg \min_{\mathbf{w}_{\mathrm{FL},n}} \left\| \mathbf{w}_{\mathrm{FL},n} - \mathbf{w}_{\mathrm{FL},n-1} \right\|_{\mathbf{Q}_n^{-1}}^2 \\ &\text{s.t. } \varepsilon\left[n\right] = \left(1 - \mu_{\mathrm{L}} \frac{\left\| \mathbf{x}_{\mathrm{L},n} \right\|_2^2}{\left\| \mathbf{x}_{\mathrm{L},n} \right\|_2^2 + \delta_{\mathrm{L}}} - \mu_{\mathrm{FL}} \frac{\left\| \mathbf{g}_n \right\|_{\mathbf{Q}_n}^2}{\left\| \mathbf{g}_n \right\|_{\mathbf{Q}_n}^2 + \delta_{\mathrm{PFL}}} \right) e\left[n\right] \end{aligned} \tag{5}$$

whose formulation is based on the derivation of a constraint relating the *a priori* error $e\left[n\right]$ with the *a posteriori* error $\varepsilon\left[n\right]$. The complete derivation of (5) can be found in [26].

In (4) and (5), μ_{L} and μ_{FL} are step-size parameters, δ_{L} and δ_{PFL} are regularization factors, and

$$\mathbf{Q}_n = \mathrm{diag}\left\{ \; q_0\left[n\right] \quad \ldots \quad q_{M_{\mathrm{e}}-1}\left[n\right] \; \right\}, \tag{6}$$

is the proportionate matrix, which can be chosen as diagonal, that performs a mask on the filter according to its sparsity. Such matrix aims at weighting the coefficients of $\mathbf{w}_{\mathrm{FL},n}$ proportionally to the contribution they provide to the nonlinear modeling. For this reason, we denote this architecture as *proportionate SFLAF* (PSFLAF). Note that when sparsity is not taken into account, $\mathbf{Q}_n = \mathbf{I}$ and the PSFLAF turns to an SFLAF model [24].

The diagonal elements of $\mathbf{Q}_n$ are computed by evaluating the nonlinear filter estimate at the previous time instant. The larger the absolute of $\mathbf{w}_{\mathrm{FL},n-1}$, the higher the corresponding weighting. Therefore, the resulting proportionate filter on the nonlinear branch converges faster than the classic FLAF when functional links show any sparseness characteristics.

For the choice of the diagonal elements of (6), we considered the improved proportionate NLMS (IPNLMS) algorithm [43], since it was extensively used with robust results in problems, such as acoustic echo cancellation, where sparsity can be described by its energy concentration [5], as also occurs in the case of functional links. However, nothing prevents us from using any other proportionate algorithm (e.g., [10, 44–46], among other). Based on the IPNLMS, the diagonal elements of $\mathbf{Q}_n$ can be calculated as:

$$q_k\left[n\right] = \frac{1-\rho}{2M_{\mathrm{e}}} + \left(1+\rho\right)\frac{\left|w_{\mathrm{FL},k}\left[n-1\right]\right|}{2\left\|\mathbf{w}_{\mathrm{FL},n-1}\right\|_1 + \xi} \tag{7}$$

with $k = 0, \ldots, M_{\mathrm{e}} - 1$ and $-1 \leq \rho \leq 1$; and ξ is a small positive constant that avoids divisions by zero.

The *asymmetry parameter* ρ balances the asymmetry in the update of coefficients, since when its value is close to 1 a high sparseness degree is assumed, while when $\rho = -1$ the adaptation follows the NLMS rule. The asymmetry parameter also affects the choice of the regularization parameter δ_{PFL} in (4), since $\delta_{\mathrm{PFL}} = \delta_{\mathrm{FL}}\left(1-\rho\right)/2M_{\mathrm{e}}$, where δ_{FL} is the regularization factor that might be used for a classic FLAF.

3. The Proposed Block-Based Combination Approach for Sparse Functional Links

In Subsection 2.2, we have seen that the sparseness of a functional link expansion may vary according to the expansion order P and to the amount

of nonlinearity introduced by a system. However, while the former condition is imposed by the algorithm designer and is a parameter *a priori* known, the latter condition is completely unknown since it depends on the system that generates a nonlinearity, which may vary in time, even very quickly. As a consequence, the sparseness of a functional link expansion may vary in time from a low degree to a higher one, and *vice versa*. Therefore, it is difficult to find a suitable configuration for the PSFLAF that provides optimal performance for any degree of nonlinearity.

In order to tackle this problem, we can think of more complex nonlinear filtering algorithms than the simple PSFLAF. To this end, we can develop filtering structures involving the adaptive combination of filters on the nonlinear branch to improve the robustness of the nonlinear modeling [47, 34, 36]. In particular, we combine IPNLMS filters having different parameter settings to exploit sparsity in functional links. Adaptive combination of IPNLMS filters was proved to provide effective results for the estimation of sparse acoustic impulse responses in echo cancellation application [32]. Indeed, when we combine such IPNLMS filters, the contribution of filter weights to the residual gradient noise is different for both the filters [32]. In particular, when the sparseness degree is negligible, the same adaptation gain may be assigned to all the filter coefficients. On the other hand, when the estimated response is sparse, the active filter coefficients are affected by large amount of gradient noise. Therefore, a scheme that combines active coefficients from the non-sparse filter with non-active coefficients from the sparse filter results in a reduction of the gradient noise.

In [32], it was proved that additional performance improvements can be reached if, instead of combining complete adaptive filters by means of a unique mixing parameter as in the basic combination of filters, we use a different mixing parameter for each group of adjacent coefficients when combining IPNLMS filters. In particular, by allowing different values for the mixing parameters and updating them by considering the minimization of the overall square error, we can keep the estimation of each block of coefficients provided by the filter that is incurring in a smaller error for that particular group of coefficients. The resulting combination is expected to incur in a smaller steady-state misalignment than any of the component filters [32]. We adopt this approach, also known as *block-based* adaptive combination, to exploit sparsity and improve the overall nonlinear modeling performance of FLAFs.

Let us consider two generic nonlinear FLAFs, where both the coefficient

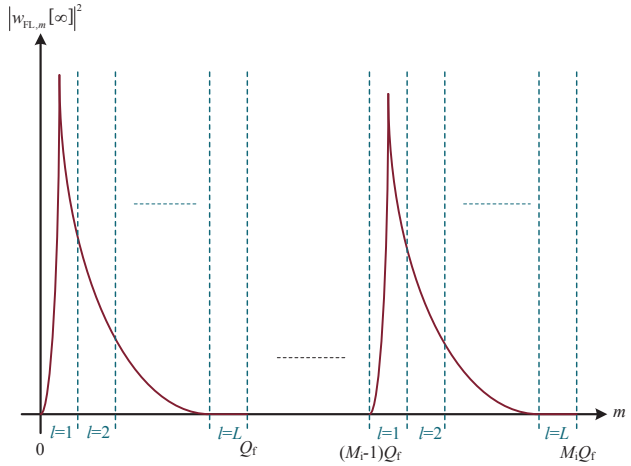


Figure 5: Expanded buffer partitioning according to the proposed block-based approach.

vectors, denoted as $\mathbf{w}_{\mathrm{FL1},n}$ and $\mathbf{w}_{\mathrm{FL2},n}$, have the same length M_e and receive the same nonlinear signal $g_n[n]$. According to the classic formulation of the block-based approach [32], we divide the coefficient vector in L blocks, each one consisting of $M_\mathrm{b} = M_\mathrm{e}/L$ coefficients. In order to avoid abrupt changes in the sparseness within each block, it is recommended to choose L such that it is a multiple of Q_f. Therefore, the output of the block-based combination can be expressed as:

$$y_\mathrm{N}[n] = \sum_{l=1}^{L} \lambda_l[n]\, y_{\mathrm{FL1}}^{(l)}[n] + (1 - \lambda_l[n])\, y_{\mathrm{FL2}}^{(l)}[n] \tag{8}$$

where $\lambda_l[n]$ is the mixing parameter associated to the l-th block and $y_{\mathrm{FL}j}^{(l)}[n]$, for $j = 1, 2$, is the output of the l-th block of the FLAF.

In this work, we propose a new block-based approach in order to take advantage of the periodic nature of the sparse energy behavior of functional links, as depicted in Fig. 3. We said in Subsection 2.2 that each group of Q_f samples (which is related to the expansion of one input sample) may be characterized by a different energy peak, but this is not relevant to the purpose of evaluating the sparseness degree of each group. Conversely, more important for that purpose is the fact that each group shows a similar energy decay to that of the adjacent groups.

This property allows to make an approximation: we may think to use the same partitioning in L blocks of a group of Q_f samples for all the M_i groups

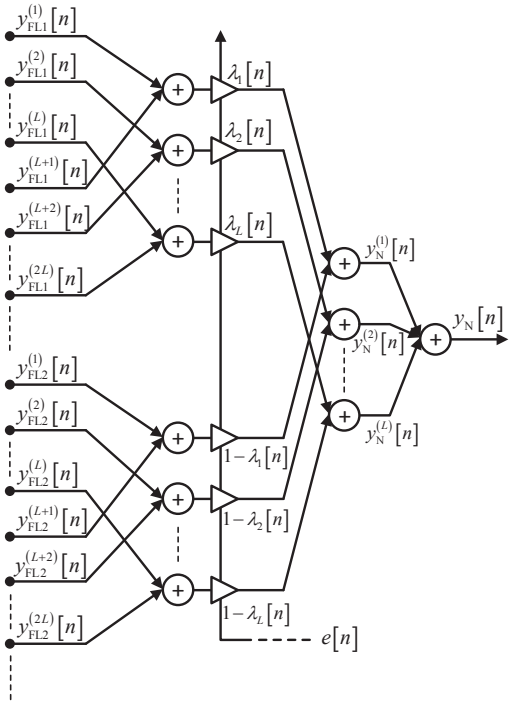


Figure 6: Scheme of the proposed block-based convex combination approach.

of Q_{f} samples. Then, we associate the l-th block of each group of Q_{f} samples, with $l = 1, \ldots, L$, with the same l-th mixing parameters. Such partitioning is depicted in Fig. 5. In this case, each blocks consists of $M_{\mathrm{b}} = Q_{\mathrm{f}}/L$ coefficients and the expanded buffer length can be also expressed as $M_{\mathrm{e}} = M_{\mathrm{b}} L M_{\mathrm{i}}$. The output of this new block-based combination is:

$$y_{\mathrm{N}}[n] = \sum_{i=0}^{M_{\mathrm{i}}-1} \left(\sum_{l=1}^{L} \lambda_l[n]\, y_{\mathrm{FL1}}^{(iL+l)}[n] + (1-\lambda_l[n])\, y_{\mathrm{FL2}}^{(iL+l)}[n] \right) \tag{9}$$

where it is possible to notice that the same L mixing parameters are used for each group of Q_{f} samples. The proposed block-based combination approach described by (9) is represented in Fig. 6.

The new block-based approach entails a computational saving of $(M_{\mathrm{i}} - 1)\, L$ adaptive mixing parameters to be computed with respect to the classical block-based approach of (8). The derivation of the adaptive mixing parameters $\lambda_l[n]$ in (9) will be presented in the next section according to the different combined PSFLAF architectures that we are going to introduce.

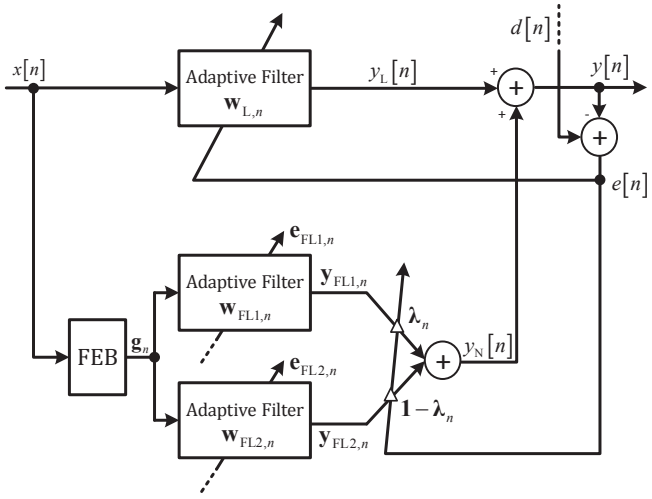


Figure 7: The combined PSFLAF #1 architecture with shared expansion.

4. Combined Proportionate SFLAF Architectures

The SFLAF in Fig. 4 is the simplest nonlinear architecture based on functional links. However, more complex structures can be thought. In this work, we propose some variants of the SFLAF, in which we leverage the combination of adaptive filters to improve the nonlinear modeling performance. In particular, the filters that we combine are all proportionate filters, therefore, the combination of such filters aims at improving performance by exploiting the sparse functional links.

4.1. Block-based cPSFLAF with shared expansion

The first architecture, denoted as *combined PSFLAF #1* (cPSFLAF#1), is depicted in Fig. 7. It is possible to notice that, similarly to the SFLAF in [24], the overall output signal results from the sum of the outputs of the linear and nonlinear branches. However, the nonlinear output is characterized by the adaptive combination between the two adaptive filters downstream of the FEB.

Such architecture allows to improve the nonlinear modeling performance by exploiting the different properties of the adaptive filters involved in the combination. This can be performed by choosing different adaptation rules for $\mathbf{w}_{\mathrm{FL1},n}$ and $\mathbf{w}_{\mathrm{FL2},n}$, or the same algorithm but using different parameter settings, e.g., different step sizes. However, the cPSFLAF#1 is mainly suited

to exploit sparsity in functional links, and this can be performed by distinguishing parameters related to the proportionate algorithms, as we are going to show in the following.

The linear branch of the cPSFLAF#1 is processed as in the SFLAF scheme, while the nonlinear branch follows the same procedure of the SFLAF until the generation of the expanded buffer $\mathbf{g}_n$. This signal is then fed into both the adaptive filters, thus generating the individual outputs and errors, for $j = 1, 2$:

$$y_{\mathrm{FL}j}[n] = \mathbf{g}_n^\top \mathbf{w}_{\mathrm{FL}j,n-1} \tag{10}$$

$$e_{\mathrm{FL}j}[n] = d[n] - (y_{\mathrm{L}}[n] + y_{\mathrm{FL}j}[n]), \tag{11}$$

where error signals (11) have been obtained following a similar procedure than in case of the combination of kernels scheme [47]. These error signals are used to adapt the filters $\mathbf{w}_{\mathrm{FL}j,n-1}$:

$$\mathbf{w}_{\mathrm{FL}j,n} = \mathbf{w}_{\mathrm{FL}j,n-1} + \mu_{\mathrm{FL}j} \frac{\mathbf{Q}_{j,n}\mathbf{g}_n}{\mathbf{g}_n^\top \mathbf{Q}_{j,n}\mathbf{g}_n + \delta_{\mathrm{PFL}j}} e_{\mathrm{FL}j}[n] \tag{12}$$

for $j = 1, 2$, where the diagonal elements of $\mathbf{Q}_{j,n}$ are computed similarly to (7), but considering that is possible to choose different parameter values for the two filters. In this way, different compromises regarding the adaptation of the nonlinear filters can be alleviated by means of combined schemes. In particular, related to the sparsity property, we could distinguish ρ_j, for $j = 1, 2$, which consequently yields $q_{j,l}[n]$ and $\delta_{\mathrm{PFL}j}$. By choosing different values for the asymmetry parameter, it is possible to exploit any sparseness degree of the nonlinear branch, increasing the robustness of the scheme with respect to the compromise imposed by the selection of parameter ρ. In fact, if we choose a high sparseness degree for the first filter, i.e., ρ_1 close to 1, and a low one for the second filter, i.e., ρ_2 close to -1, it is possible to give robustness to the model with respect to any sparseness degree of a functional link expansion.

As it is possible to see in Fig. 7, the overall output of the nonlinear branch is achieved by combining convexly the individual filter outputs (10) in a block-based fashion, following the approach described in Subsection 3 and depicted in Fig. 6. It is worth noting that in Fig. 7 the convex combination on the nonlinear branch is performed by using the scheme of Fig. 6, but it is simplified for clarity, being $\mathbf{y}_{\mathrm{FL}j,n} = \left[\begin{array}{ccc} y_{\mathrm{FL}j}^{(1)}[n] & \dots & y_{\mathrm{FL}j}^{(L)}[n] \end{array} \right]^\top$ and $\boldsymbol{\lambda}_n = \left[\begin{array}{ccc} \lambda_1[n] & \dots & \lambda_L[n] \end{array} \right]^\top$, with $j = 1, 2$.

Since the cPSFLAF#1 involves a shared functional link expansion, the input signal to the filters $\mathbf{w}_{\mathrm{FL}j,n-1}$ is the same. Hence, the output equation for $y_{\mathrm{N}}[n]$ can be expressed as:

$$y_{\mathrm{N}}[n] = \sum_{i=0}^{M_{\mathrm{i}}-1} \left(\sum_{l=1}^{L} \lambda_l[n]\, \mathbf{g}_n^{(iL+l)\top} \mathbf{w}_{\mathrm{FL1},n-1}^{(iL+l)} + (1-\lambda_l[n])\, \mathbf{g}_n^{(iL+l)\top} \mathbf{w}_{\mathrm{FL2},n-1}^{(iL+l)} \right) \tag{13}$$

where, in this case, the product $\mathbf{g}_n^{(iL+l)\top} \mathbf{w}_{\mathrm{FL}j,n-1}^{(iL+l)}$ (with $j = 1, 2$) yields the output of the l-th block $y_{\mathrm{FL}j}^{(iL+l)}[n]$, used for the block-based combination as depicted in Fig. 6.

The adaptive mixing parameter $\lambda_l[n]$ in (13) balances the combination between the l-th blocks of each group of Q_{f} samples of the two filters $\mathbf{w}_{\mathrm{FL}j}[n]$ ($j = 1, 2$), giving more importance to the best performing filter [30]. Such awareness is obtained according to a mean square error minimization. In particular, the adaptation of $\lambda_l[n]$ aims at minimizing the power of the global error signal $e[n]$ and it is performed by using an auxiliary adaptive parameter for each block $a_l[n]$, which is related to $\lambda_l[n]$ by means of a sigmoidal function that keeps the mixing parameter in the range $[0, 1]$, and defined according to [48, 33] as:

$$\lambda_l[n] = \beta \left(\frac{1}{1 + e^{-a_l[n]}} - \alpha \right), \tag{14}$$

where $\alpha = 1/(1 + e^4)$ and $\beta = 1/(1 - 2\alpha)$. The auxiliary parameter for the l-th block is derived by a gradient descent rule:

$$a_l[n] = a_l[n-1] + \frac{\mu_{\mathrm{c}}}{\beta r_l[n-1]} e[n]\, \Delta y_l[n]\, (\lambda_l[n] + \alpha\beta)\, (\beta - \alpha\beta - \lambda_l[n]) \tag{15}$$

where

$$\Delta y_l[n] = \sum_{i=0}^{M_{\mathrm{i}}-1} \mathbf{g}_n^{(iL+l)\top} \left(\mathbf{w}_{\mathrm{FL1},n-1}^{(iL+l)} - \mathbf{w}_{\mathrm{FL2},n-1}^{(iL+l)} \right) \tag{16}$$

In (15), μ_{c} is the step-size parameter of the adaptive combination, $r_l[n] = \gamma r_l[n-1] + (1-\gamma)\, \Delta y_l^2[n]$ is the estimated power of $\Delta y_l[n]$ that permits a normalized adaptation of $a_l[n]$, and γ is a smoothing factor.

The overall error signal $e[n]$ is derived as:

$$e[n] = d[n] - (y_{\mathrm{L}}[n] + y_{\mathrm{N}}[n]) \tag{17}$$

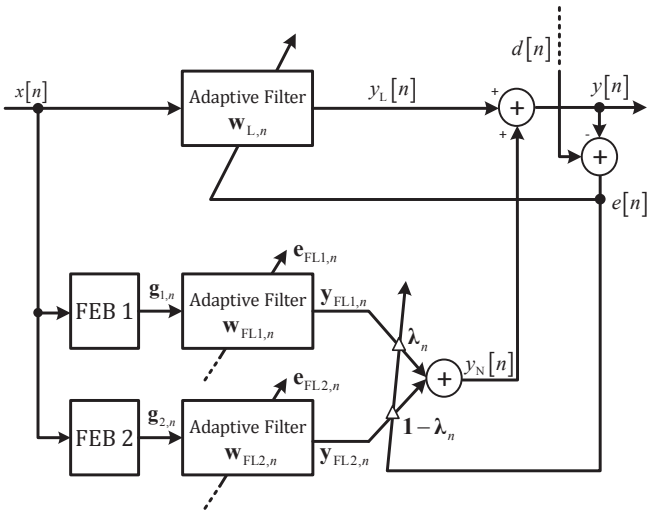


Figure 8: The combined PSFLAF #2 architecture involving different FEBs.

and it is used to update both the linear filter $\mathbf{w}_{\mathrm{L},n}$ and each auxiliary parameter $a_l\,[n]$.

As said in Subsection 2.2, the sparseness of functional links may vary according to the expansion order P and to the nonlinearity level. The cPSFLAF#1 is well suited for the latter case since it exploits the different settings of the proportionate filters and uses the same P for both the combined filters. Therefore, the cPSFLAF#1 architecture represents the best solution when the nonlinearity varies its level in time. The former case will be addressed in the next subsection.

4.2. Block-based cPSFLAF with different FEBs

We propose another nonlinear architecture in which the different sparseness degrees of the two combined proportionate filters are achieved using same proportionaly factors but different expansion orders. Such architecture is denoted as *combined PSFLAF #2* (cPSFLAF#2) and it is depicted in Fig. 8. As it is possible to see, the peculiarity of the cPSFLAF#2 is the presence of two different FEBs on the nonlinear branch.

Two FEBs with different expansion orders $P_{\mathrm{FL}j}$, with $j = 1, 2$, imply two different expansion vectors $\mathbf{g}_{j,n}$ and two proportionate filters $\mathbf{w}_{\mathrm{FL}j}$ having different lengths $M_{\mathrm{e}j}$. In particular, to simplify the notation, we choose $P_{\mathrm{FL1}} < P_{\mathrm{FL2}}$, so that $M_{\mathrm{e1}} < M_{\mathrm{e2}}$. The output of each proportionate filter on

the nonlinear branch of the cPSFLAF#2 can be expressed as:

$$y_{\mathrm{FL}j}\left[n\right] = \mathbf{g}_{j,n}^{\top}\mathbf{w}_{\mathrm{FL}j,n-1} \tag{18}$$

for $j = 1, 2$. The error signals $e_{\mathrm{FL}j}\left[n\right]$, instead, keep the same expression of (11). The update equation of the two proportionate filters involves the different expansion buffers and becomes:

$$\mathbf{w}_{\mathrm{FL}j,n} = \mathbf{w}_{\mathrm{FL}j,n-1} + \mu_{\mathrm{FL}j}\frac{\mathbf{Q}_{j,n}\mathbf{g}_{j,n}}{\mathbf{g}_{n}^{\top}\mathbf{Q}_{j,n}\mathbf{g}_{j,n} + \delta_{\mathrm{PFL}j}}e_{\mathrm{FL}j}\left[n\right]. \tag{19}$$

The diagonal elements of $\mathbf{Q}_{j,n}$ are computed as for the cPSFLAF#1.

The two proportionate filters on the nonlinear branch are combined convexly, as represented in the simplified scheme of Fig. 8. Even in this case, similarly to the cPSFLAF#1, the convex combination is performed by applying the block-based approach of Fig. 6. Due to the different length of the two filters, the block-based approach is slightly different from the cPSFLAF#1. In this case, the number of functional links changes for the two FEBs, being $Q_{\mathrm{f1}} = 2P_{\mathrm{FL1}}$ and $Q_{\mathrm{f2}} = 2P_{\mathrm{FL2}}$, so that $M_{ej} = Q_{\mathrm{f}j}M_{\mathrm{i}}$, with $j = 1, 2$. We choose a number of blocks L that is divisible by the larger number of functional links for the sets used by the two FEBs, i.e., Q_{f2}, so that each block consists of $M_{\mathrm{b}} = Q_{\mathrm{f2}}/L$ coefficients. However, since we choose $Q_{\mathrm{f1}} < Q_{\mathrm{f2}}$, we may have null blocks for the first expansion. For his reason, we define $L_{\mathrm{NZ}} = \lceil Q_{\mathrm{f1}}/M_{\mathrm{b}} \rceil$ as the number of non-zero blocks for the first expansion. We must also consider that the last block may be shorter than M_{b} coefficients, so we pad the last block with a number of zeros equal to Q_{f1} mod M_{b}. Therefore, the output of the nonlinear branch of the cPSFLAF#2 is:

$$y_{\mathrm{N}}\left[n\right] = \sum_{i=0}^{M_{\mathrm{i}}-1}\left(\sum_{l=1}^{L_{\mathrm{NZ}}}\lambda_{l}\left[n\right]\mathbf{g}_{1,n}^{(iL+l)\top}\mathbf{w}_{\mathrm{FL1},n-1}^{(iL+l)} + \sum_{l=1}^{L}\left(1-\lambda_{l}\left[n\right]\right)\mathbf{g}_{2,n}^{(iL+l)\top}\mathbf{w}_{\mathrm{FL2},n-1}^{(iL+l)}\right) \tag{20}$$

where it can be seen that, differently from the cPSFLAF#1 output in (13), in this case $y_{\mathrm{FL}j}^{(iL+l)}\left[n\right] = \mathbf{g}_{j,n}^{(iL+l)\top}\mathbf{w}_{\mathrm{FL}j,n-1}^{(iL+l)}$, for $j = 1, 2$. It is also worth noting in (20) that for the first L_{NZ} blocks there is a convex combination between the two filters, while for the $L - L_{\mathrm{NZ}}$ remaining blocks the combination involves only the second filter $\mathbf{w}_{\mathrm{FL2},n}$, and this can be also interpreted as a convex combination with null virtual blocks, called all-zero blocks [38]. The representation of the energy of the sum of the blocks for the two filters $\mathbf{w}_{\mathrm{FL1}}$ and $\mathbf{w}_{\mathrm{FL2}}$ is depicted in Fig. 9.

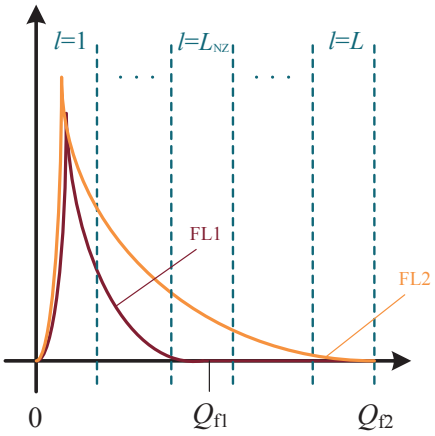


Figure 9: Representation of the energy of the L resulting blocks for the two filters $\mathbf{w}_{\mathrm{FL1}}$ and $\mathbf{w}_{\mathrm{FL2}}$.

All the mixing parameters $\lambda_l[n]$ and the auxiliary parameters $a_l[n]$, with $l = 1, \ldots, L$, are computed respectively by applying the same equations (14) and (15) used for the cPSFLAF#1. However, in this case $\Delta y_l[n]$, in the update equation (15) of the l-th auxiliary parameter, is computed as:

$$\Delta y_l[n] = \begin{cases} \sum_{i=0}^{M_{\mathrm{i}}-1} \mathbf{g}_{1,n}^{(iL+l)\top} \mathbf{w}_{\mathrm{FL1},n-1}^{(iL+l)} - \mathbf{g}_{2,n}^{(iL+l)\top} \mathbf{w}_{\mathrm{FL2},n-1}^{(iL+l)}, & l = 1, \ldots, L_{\mathrm{NZ}} \\ -\sum_{i=0}^{M_{\mathrm{i}}-1} \mathbf{g}_{2,n}^{(iL+l)\top} \mathbf{w}_{\mathrm{FL2},n-1}^{(iL+l)}, & l = L_{\mathrm{NZ}}+1, \ldots, L. \end{cases} \tag{21}$$

Once computed the output $y_{\mathrm{N}}[n]$ of the nonlinear branch, we can achieve the overall error signal $e[n]$, which updates both $\mathbf{w}_{\mathrm{L},n}$ and each $a_l[n]$, for $l = 1, \ldots, L$, by using (17).

Due to its characteristics, the cPSFLAF#2 is well suited in those cases where the nonlinearity is nearly static in time, but its level is unknown *a priori*.

4.3. Block-based Hierarchical cPSFLAF

In the previous subsections, we have introduced two nonlinear filtering architectures that exploit sparse functional links when the nonlinearity to be modeled is changing in time (cPSFLAF#1) and also when it is nearly static (cPSFLAF#2). However, if we do not have enough information about the behavior of the nonlinearity in time, we can think to generalize the previous architectures. To this end, we introduce a *hierarchical combined PSFLAF*

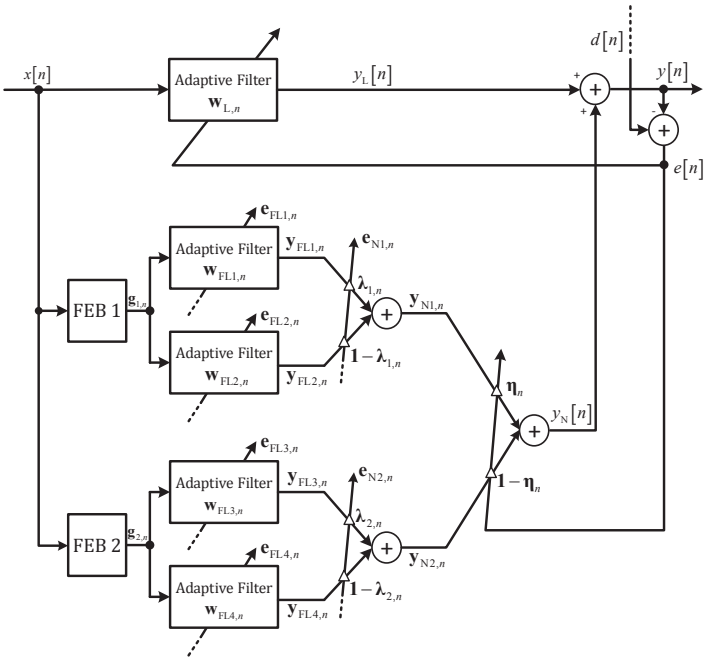


Figure 10: The hierarchical combined PSFLAF architecture.

(HcPSFLAF) model, depicted in Fig. 10, which is an adaptive nonlinear architecture able to automatically select the constituents that mostly contribute to the nonlinear modeling.

The HcPSFLAF involves a first stage of combinations, in which two pairs of proportionate filters are combined after a FEB. We can distinguish the 4 filters involved in the combinations according to the asymmetry parameters (or to the step-size parameters as an alternative) and to the expansion orders. In particular, the filters of the first combination, i.e., $\mathbf{w}_{\mathrm{FL1},n}$ and $\mathbf{w}_{\mathrm{FL2},n}$, have different asymmetry parameters, respectively $\rho_{\mathrm{A}} \to 1$ and $\rho_{\mathrm{B}} \to -1$, and the same small value for the expansion order P_{A}. The second pair of first-stage combined filters, $\mathbf{w}_{\mathrm{FL3},n}$ and $\mathbf{w}_{\mathrm{FL4},n}$, keep the same asymmetry parameters of the first pair, i.e., $\rho_{\mathrm{A}} \to 1$ and $\rho_{\mathrm{B}} \to -1$ respectively, but both filters share a larger value for the expansion order $P_{\mathrm{B}} > P_{\mathrm{A}}$.

Both the convex combinations on the first stage will be achieved by adopting the block-based convex combination used for the cPSFLAF#1 and described by (13), since each pair of filter shares the same expansion. The outputs of the first-stage combinations, i.e., $y_{\mathrm{N1}}[n]$ and $y_{\mathrm{N2}}[n]$, are in turn

convexly combined in a second stage in order to yield the overall output of the nonlinear branch $y_{\mathrm{N}}[n]$, as represented in Fig. 10, where the l-th mixing parameter of the second stage is denoted with $\eta_l[n]$. Even this second-stage combination involves a block-based approach. However, unlike the first-stage combinations, here different expansion orders are involved, i.e., P_{A} and P_{B}, hence the output $y_{\mathrm{N}}[n]$ is achieved by using the block-based convex combination used for the cPSFLAF#2 and described by (20). The overall error signal is obtained as usual by (17) and it is used to adapt the adaptive filter on the linear path and the mixing parameters on the second stage.

4.4. Computational Analysis

In order to choose the best suitable combined PSFLAF architecture for a certain problem, it is fundamental to be aware of its computational cost. The complexity of a PSFLAF-based architecture may depend on several parameters, including the parameters of the adaptive algorithms and the ones related to the adaptive combinations. All the proposed combined architectures are evolutions of the PSFLAF, which therefore represents the baseline model even from a computational point of view. Based on the analysis in [26], the PSFLAF requires $3M+2$ multiplications and $3M-1$ additions for the linear part and $12PM_{\mathrm{i}}+P+3$ multiplications, $8PM_{\mathrm{i}}+1$ additions and $2P$ function evaluations for the nonlinear branch, considering an iterative buffer shift for the input vector to the FEB.

With respect to the PSFLAF, the cPSFLAF#1 architecture involves an additional cost for the adaptation of the second IPNLMS and the block-based combination on the nonlinear branch, thus leading to an overhead of $12PM_{\mathrm{i}}+13L+6$ multiplications, $6PM_{\mathrm{i}}+(3M_{\mathrm{i}}+8)L+1$ additions and L exponentiations. It is worth noting that $L \ll 2PM_{\mathrm{i}}$ and therefore, the block-based adaptive combination does not add a significant computational overload with respect to the number of products required by the adaptation of the filters. The cPSFLAF#2 architecture is a little bit more complex than cPSFLAF#1 since it involves an additional FEB. However, in this case, we slightly different parameters, including two expansion orders, P_{FL1} and P_{FL2}, and the number of non-zero blocks L_{NZ}. The overall complexity of the cPSFLAF#2 is given by $12\left(P_{\mathrm{FL1}}+P_{\mathrm{FL2}}\right)M_{\mathrm{i}}+P_{\mathrm{FL2}}+L_{\mathrm{NZ}}+9L+9$ multiplications, $8\left(P_{\mathrm{FL1}}+P_{\mathrm{FL2}}\right)M_{\mathrm{i}}+2L_{\mathrm{NZ}}M_{\mathrm{i}}+LM_{\mathrm{i}}+7L+1$ additions, $2P_{\mathrm{FL2}}$ function evaluations and $2L$ exponentiations. If we consider that $P_{\mathrm{FL1}}+P_{\mathrm{FL2}} > P$ and $L_{\mathrm{NZ}} < L$, the most significant computational overhead of the cPSFLAF#2 with respect to the cPSFLAF#1 can be expressed in terms of $12\left(P-\frac{P_{\mathrm{FL1}}+P_{\mathrm{FL2}}}{2}\right)M_{\mathrm{i}}$

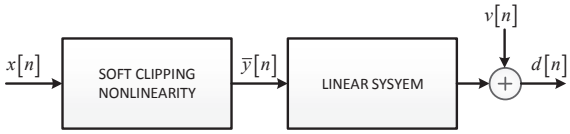


Figure 11: Nonlinear system to be identified.

multiplications, $2\left(P-\frac{P_{\text{FL1}}+P_{\text{FL2}}}{2}\right)M_{\text{i}}$ additions and $P_{\text{FL2}}-P$ function evaluations.

Excluding the linear branch, which is the same of the other architectures, the nonlinear part of hierarchical cPSFLAF architecture involves the double of the complexity of the nonlinear branch of the cPSFLAF#1 with the addition of a second stage of block-based combinations. In details, the nonlinear branch of the HcPSFLAF requires $24\left(P_{\text{FL1}}+P_{\text{FL2}}\right)M_{\text{i}}+P_{\text{FL2}}+L_{\text{NZ}}+35L+15$ multiplications, $16\left(P_{\text{FL1}}+P_{\text{FL2}}\right)M_{\text{i}}+2L_{\text{NZ}}M_{\text{i}}+7LM_{\text{i}}+23L+4$ additions, $2P_{\text{FL2}}$ function evaluations and $3L$ exponentiations. In terms of multiplications, the complexity of the nonlinear branch of the HcPSFLAF can be roughly considered as the double of the complexity of the nonlinear branch of the cPSFLAF#2.

In general, as expected, the computational complexity of the PSFLAF-based architectures increases with the addition of new processing blocks in the architecture, until arriving to the HcPSFLAF which can be also expanded by adding further stages of combination. The HcPSFLAF always generalizes the cPSFLAF#1 and cPSFLAF#2 architectures, but in order to keep contained the computational complexity it is always preferable, when possible, to choose one of the two basic schemes. Moreover, the computational complexity may vary significantly according to the parameter setting, therefore, an accurate choice of the parameters may also help in saving computational resources.

5. Experimental Results

In this section, we assess the effectiveness of the proposed combined architectures and we show their capability of exploiting sparsity. For all the experiments, the system to be identified has a Hammerstein structure, i.e., it involves a nonlinear block followed by a linear one, as depicted in Fig. 11. The input signal $x\left[n\right]$ is distorted by a soft-clipping nonlinearity as follows

[26]:

$$\overline{y}\left[n\right] = \begin{cases} \frac{2}{3\zeta} x\left[n\right] & , \quad 0 \leq \left|x\left[n\right]\right| \leq \zeta \\ \operatorname{sign}\left(x\left[n\right]\right) \frac{3-\left(2-\left|x[n]\right|/\zeta\right)^2}{3} & , \quad \zeta \leq \left|x\left[n\right]\right| \leq 2\zeta \\ \operatorname{sign}\left(x\left[n\right]\right) & , \quad 2\zeta \leq \left|x\left[n\right]\right| \leq 1 \end{cases} \tag{22}$$

where $0 < \zeta \leq 0.5$ is a nonlinearity threshold. Such threshold can be appropriately chosen to introduce strong (ζ close to 0.5) or mild (ζ close to 0) nonlinearity in order to highlight the sparsity behavior of the functional link expansions with respect to the nonlinear system to be identified, according to Subsection 2.2. The signal $\overline{y}\left[n\right]$ is then fed into a linear system, formed by $M = 10$ independent random values between -1 and 1. An independent and identically distributed (i.i.d.) noise signal $v\left[n\right]$ is added at output of the whole plant, in order to provide 20 dB of signal-to-noise ratio (SNR). The input signal $x\left[n\right]$ is generated by means of a first-order autoregressive model, whose transfer function is $\sqrt{1-\theta^2}/\left(1-\theta z^{-1}\right)$, with $\theta = 0.8$, fed with an i.i.d. Gaussian random process. Performance is evaluated in terms of the *excess mean-square error* (EMSE), expressed in dB as $\mathrm{EMSE}\left[n\right] = 10\log_{10}\left(\mathrm{E}\left\{\left(e\left[n\right]-v\left[n\right]\right)^2\right\}\right)$, which is averaged over 1000 runs with respect to input and noise. In order to facilitate the visualization, curves are also smoothed by a moving-average filter.

5.1. Evaluation of the cPSFLAF#1

The combined architecture adopted for the cPSFLAF#1 can lead to performance improvement by exploiting the properties of the adaptive filters in combination on the nonlinear branch, e.g., by using different parameter setting [34, 36]. Here, we want to exploit the sparseness of functional links and this can be performed by distinguish the peculiar parameter of the IPNLMS, i.e., the asymmetry parameter. In particular, we set $\rho_1 = 0.9$ and $\rho_2 = -1$ in order to specialize the architecture regardless of any sparseness degree. It should be noted that we chose a value of ρ_1 a little bit lower than 1 and we select ρ_2 to be exactly equal to -1 (since this leads to an NLMS adaptation instead of an IPNLMS scheme), thus yielding very similar performance behavior but with reduced computational cost.

The rest of the parameter setting for the two filters on the nonlinear path involves the same values for the step sizes $\mu_{\mathrm{FL1}} = \mu_{\mathrm{FL2}} = 0.1$, the regularization factors $\delta_{\mathrm{FL1}} = \delta_{\mathrm{FL2}} = 10^{-3}$, memoryless trigonometric expansions with order $P = 20$ and, as input buffer length, the same length of the linear system to be identified and of the coefficient vector $\mathbf{w}_{\mathrm{L}}$, i.e, $M_{\mathrm{i}} = M$. The other

parameters of the filter $\mathbf{w}_{\mathrm{L}}$, adapted by an NLMS algorithm, are chosen as $\mu_{\mathrm{L}} = 0.1$, $\delta_{\mathrm{L}} = 10^{-3}$. Each block-based combination is adapted by using the same step-size value $\mu_{\mathrm{c}} = 0.1$, smoothing factor $\gamma = 0.9$ and initial values $a_l[0] = 0$ and $r_l[0] = 1$. The length of the experiments is 40000 samples.

First, we want to evaluate the effectiveness of using blockwise adaptive combination in the proposed scheme. Since such strategy aims at reducing the steady-state EMSE of the combined scheme, we investigate how the block size affects the steady-state performance. Since we want to evaluate the sparseness of the functional link expansion with respect to the nonlinear characteristics of the unknown system according to Subsection 2.2, we need to set the clipping threshold of (22) appropriately. For this experiment, we choose a strong nonlinearity level by setting $\zeta = 0.03$. Results are depicted in Fig. 12, where the values of the steady-state EMSE are shown on varying the number of blocks from $L = 1$, which implies the adaptation of a full block with size $M_{\mathrm{b}} = Q_{\mathrm{f}} = 40$, to $L = 20$, which implies the adaptation of blocks with size $M_{\mathrm{b}} = Q_{\mathrm{f}}/L = 2$. In the figure, the results from individual PSFLAF schemes are reported as a comparison. First, we may note that the use of a blockwise combination, obtained by choosing $L > 1$ always produces better results with respect to the full-block combination, i.e., the strategy originally adopted in [36]. However, it is also possible to notice that the combined architecture is not very sensitive to the choice of the block number L, since an ERLE reduction of about 2 dB occurs between the best choice $L = 5$ and the worst case $L = 2$. We may also note that reducing too much the block size leads to a performance decrease, consequence of the larger gradient noise power in the adaptation of blocks with reduced lengths [38], besides increasing the computational cost. Overall, we can conclude that suitable choices for the number of blocks are provided by intermediate values, i.e., for $L = 4, \ldots, 10$.

We want also to assess the convergence performance of the cPSFLAF#1, with respect to individual PSFLAF schemes, i.e., those providing the same filter on the linear branch but only one filter on the nonlinear branch. We also add the behavior of a linear NLMS filter, to provide an idea of how strong is the nonlinearity to be modeled. In order to produce a change in the nonlinearity, we choose a clipping threshold $\zeta = 0.08$ for the first half of the experiment, while we set $\zeta = 0.05$ that leads to a stronger level of nonlinearity for the second half. Results are depicted in Fig. 13(a), where it is possible to note how the cPSFLAF#1 architectures outperform individual PSFLAFs. However, the cPSFLAF#1 with a number of blocks $L = 5$ achieves the

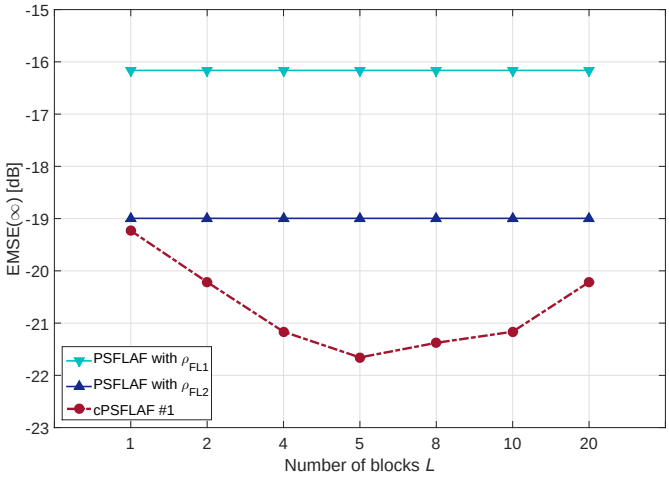


Figure 12: Steady-state EMSE of the cPSFLAF#1 on varying the number of blocks in the combination.

best performance even with respect to the cPSFLAF#1 with a single block ($L = 1$), which is the method previously proposed in [36]. In particular, the combined architectures take advantage of the convergence performance of the PSFLAF with the highest asymmetry parameter (i.e., ρ_1) in the first half of the experiment where the high expansion order with respect to the nonlinearity introduced by the system generates a very sparse behavior of functional links. On the other hand, in the second half of the experiment, the combined schemes exploit the fast convergence rate of the non-proportionate SFLAF (i.e., the PSFLAF with $\rho_2 = -1$), and the steady-state precision of the PSFLAF with ρ_1. This performance improvement can be explained by the mixing parameter evolution, depicted in Fig. 13(b) for the block $l = 1$. In particular, in the first half of the experiment the cPSFLAF#1 algorithm adaptively selects PSFLAF with ρ_1. On the other hand, in the second half it first selects the PSFLAF with ρ_2, since it mostly includes active coefficients and benefits from the smaller gradient noise achieved by the the PSFLAF with the smallest value of the asymmetric parameter. Then, after a while, the block includes some non-active coefficients and the mixing parameter switches to the PSFLAF with ρ_1. Fig. 12 and 13 expose the advantage of considering a blockwise combination both in convergence and steady-state phases.

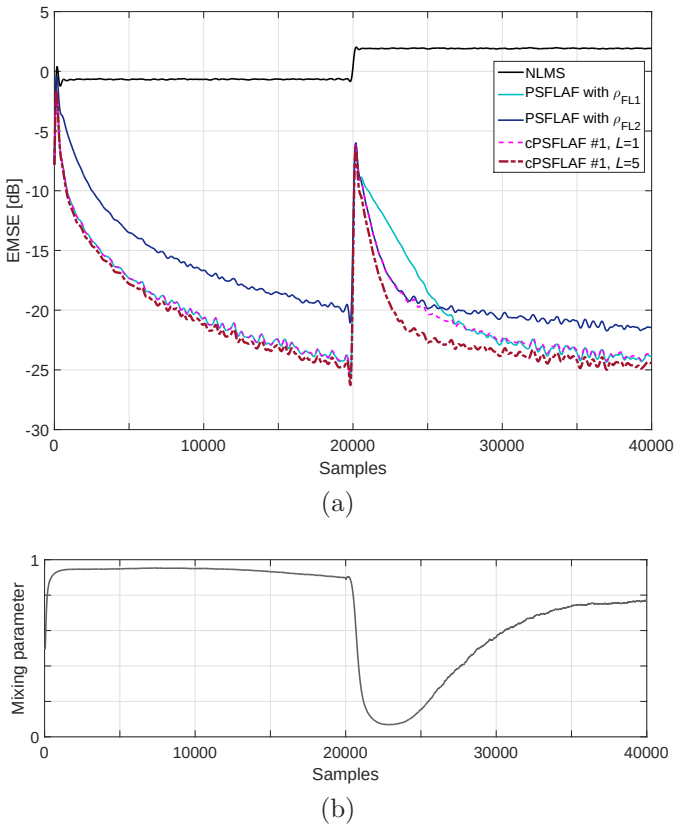


Figure 13: Performance behavior in terms of the ERLE of the cPSFLAF#1 with shared expansion (a) and the related mixing parameter evolution for the first block in case CPSFLAF#1, with $L = 5$ (b).

5.2. Evaluation of the cPSFLAF#2

In order to assess the effectiveness of the cPSFLAF#2, we consider a scenario with a nearly static nonlinearity for the whole length of the experiment, according to what explained in Subsection 4.2. For this reason, we keep the clipping threshold fixed to the value $\zeta = 0.08$. The other factor that mainly affects the sparseness of the functional link expansion according to the analysis introduced in Subsection 2.2 is the expansion order. For this reason, in this case we exploit the different sparseness degree of the two filters on the nonlinear branch by choosing different expansion orders and the same asymmetry parameter. In particular, we set the expansion orders as $P_{\text{FL1}} = 5$ and $P_{\text{FL2}} = 20$, and the asymmetry parameters as $\rho_1 = \rho_2 = 0$, which is an intermediate value. The rest of the parameter setting is the same

of the previous set of experiments.

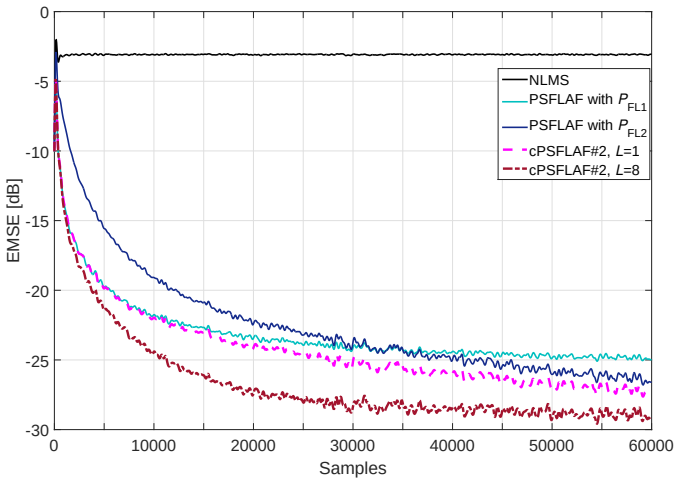
In Fig. 14(a), results in terms of the EMSE are depicted for the cPSFLAF#2 architecture using both the single-block strategy [36] (i.e., with $L = 1$) and the best block-based strategy (achieved by setting $L = 8$). We compare these results again with the individual PSFLAFs and with the NLMS filter. It is worth noting that the PSFLAF with the small expansion order, i.e., P_{FL1}, provides faster convergence rate than the PSFLAF with P_{FL2}, which, on the other hand, shows better precision at steady state. The cPSFLAF#2 scheme with single-block strategy (i.e., $L = 1$) exploits the capabilities of the two PSFLAFs providing always better performance. However, such result is further improved by the proposed cPSFLAF#2 with block-based strategy, thus showing a significant gain over the same architecture with single-block strategy.

We also show in Fig. 14(b) the behavior of all the mixing parameter of the cPSFLAF#2 with $L = 8$. It can be noted how blocks may behave even more differently from each others. In particular, first blocks are adapted with both schemes, especially when they reach convergence. The adaptation of intermediate blocks is mainly performed by the PSFLAF with P_{FL2} (i.e., their mixing parameters tend to 0) since there are energy in this components, and the first scheme has no enough coefficients for these blocks (i.e., it is too short). Finally, the first and the last blocks are mainly adapted by the PSFLAF with P_{FL1}, as their mixing parameters are more close to 1. The last blocks converge to the first scheme because there is no energy in these components, and the combinations converge to the all-zero blocks of the first scheme.

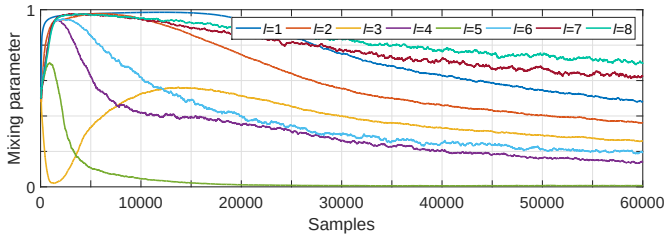
5.3. Evaluation of the Hierarchical cPSFLAF

In the last set of experiments, we evaluate the performance of the HcPSFLAF. We consider again the nonlinear system identification problem represented in Fig. 11 including a change of the nonlinearity degree at half of the experiment length. In particular, for the first part of the experiment we introduce a mild nonlinearity by setting the clipping threshold to $\zeta = 0.25$, while for the second half of the experiment we increase the nonlinearity degree by setting $\zeta = 0.15$.

The hierarchical structure of the HcPSFLAF allows to specialize the learning not only with respect to one parameter, e.g., the asymmetry parameter for the cPSFLAF#1 and the expansion order for the cPSFLAF#2, but differentiating more than one parameters. Due to the two stages of



(a)

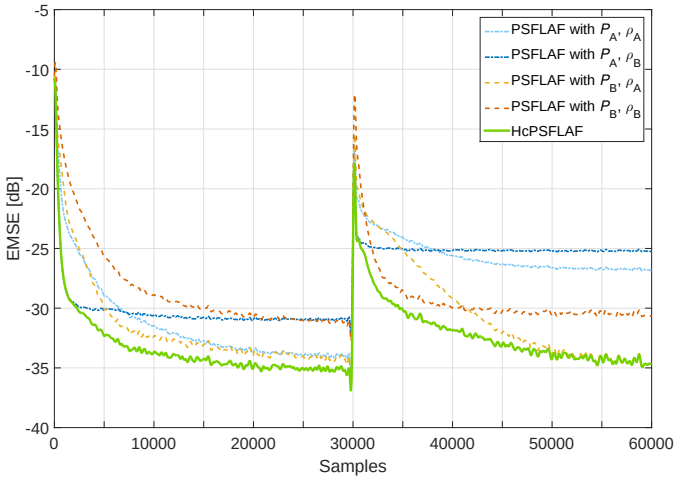


(b)

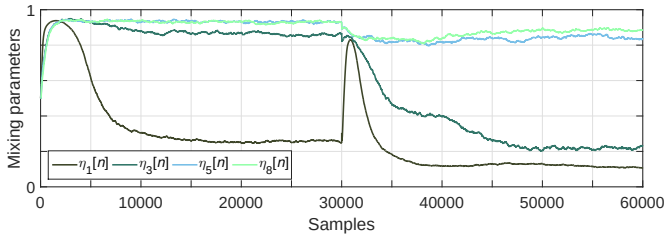
Figure 14: Performance behavior in terms of the ERLE of the cPSFLAF#2 with different expansions (a) and the related mixing parameter evolutions (b).

combinations, we may consider two different values for both the asymmetry parameter and the expansion order. In particular, we choose a high asymmetry parameter $\rho_{\mathrm{A}} = 0.9$ for the first and the third filters of the nonlinear branch, i.e., $\mathbf{w}_{\mathrm{FL1},n}$ and $\mathbf{w}_{\mathrm{FL3},n}$, and $\rho_{\mathrm{B}} = -1$ for the other two filters of the nonlinear branch. On the other hand, we choose a small expansion order $P_{\mathrm{A}} = 3$ for $\mathbf{w}_{\mathrm{FL1},n}$ and $\mathbf{w}_{\mathrm{FL2},n}$ and a larger order $P_{\mathrm{B}} = 30$ for $\mathbf{w}_{\mathrm{FL3},n}$ and $\mathbf{w}_{\mathrm{FL4},n}$. The number of blocks considered for the block-based adaptation of the HcPSFLAF is $L = 8$. The rest of the parameter setting is the same used for the previous set of experiments.

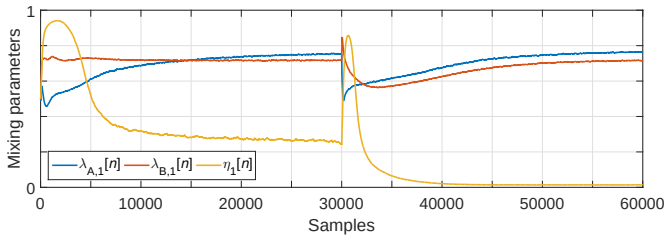
Results in terms of the EMSE are depicted in Fig. 15(a), where it is possible to notice the behavior of the four different PSFLAFs and how the HcPSFLAF exploits the capabilities of all the filters, thus behaving always



(a)



(b)



(c)

Figure 15: Performance behavior in terms of the ERLE of the HcPSFLAF (a), with the related mixing parameter evolutions for the most representative blocks of the second-stage combination (b), and mixing parameter evolutions of the first block for the first and second stage of combinations (c).

as the best filter and, very often, even better than the best individual filter thanks to the blockwise combination. We also show the evolution of the most representative mixing parameters for the blocks of the second-stage

combination in Fig. 15(b), and the evolution of the mixing parameters for the first block of the first and second stage of combination in Fig. 15(c). From these figures, it is possible to gather how the HcPSFLAF exploits sparsity of functional links. For instance, we can see that the first block of the HcPSFLAF is adapted by exploiting the filters with the most sparse features both in the first and in the second combination. This shows how each block can affect differently the adaptation and improve the overall modeling performance.

6. Conclusions

In this paper, new nonlinear filtering architectures have been proposed that take advantage of any sparseness degree of functional link expansions. The proposed models are characterized by an adaptive combination of two proportionate filters on the nonlinear branch that yields an improvement of the overall modeling performance. The adaptive combination is performed by a new block-based strategy that exploits the periodic sparse nature of a functional link expansion. A hierarchical architecture has been also proposed that generalizes the two combined schemes, thus not requiring any *a priori* information about the nature of the nonlinearity to be modeled. Future research will include the combination of filters also on the linear branch and the application to nonlinear acoustic echo cancellation problems.

Acknowledgements

The work of M. Scarpiniti and A. Uncini is partially supported by the Italian National Project "GAUChO - A Green Adaptive Fog Computing and Networking Architecture", under grant number 2015YPXH4W.

The work of L. A. Azpicueta-Ruiz is partially supported by Comunidad de Madrid under grant 'CASI-CAM-CM' (id. S2013/ICE-2845), by the Spanish Ministry of Economy and Competitiveness (under grant DAMA (TIN2015-70308-REDT) and grant TEC2014-52289-R), and by the European Union.

The work of J. Arenas-García has been partly funded by MINECO project TEC2014-52289-R, and by Comunidad de Madrid project PRICAM S2013/ICE-2933.

References

[1] M. J. Fadili, J.-L. Starck, J. Bobin, Y. Moudden, Image decomposition and separation using sparse representations: An overview, Proceedings of the IEEE 98 (6) (2010) 983–994.

[2] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[3] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, M. E. Davies, Sparse representations in audio and music: From coding to source separation, Proceedings of the IEEE 98 (6) (2010) 995–1005.

[4] J. F. Gemmeke, T. Virtanen, A. Hurmalainen, Exemplar-based sparse representations for noise robust automatic speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 19 (7) (2011) 2067–2080.

[5] Y. Huang, J. Benesty, J. Chen, Acoustic MIMO Signal Processing, Signals and Communication Technology, Springer-Verlag Berlin Heidelberg, 2006.

[6] N. Kalouptsidis, G. Mileounis, B. Babadi, V. Tarokh, Adaptive algorithms for sparse system identification, Signal Procesing 91 (8) (2011) 1910–1919.

[7] M. Masood, L. H. Afify, T. Y. Al-Naffouri, Efficient coordinated recovery of sparse channels in massive MIMO, IEEE Transactions on Signal Processing 63 (1) (2015) 104–118.

[8] S. L. Gay, Dynamically regularized fast RLS with application to echo cancellation, in: IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP), Vol. 2, Atlanta, GA, 1996, pp. 957–960.

[9] D. Duttweiler, Proportionate normalized least-mean-squares adaptation in echo cancelers, IEEE Transactions on Speech and Audio Processing 8 (5) (2000) 508–518.

[10] P. A. Naylor, J. Cui, M. Brookes, Adaptive algorithms for sparse echo cancellation, Signal Procesing 86 (6) (2004) 1182–1192.

[11] T. van Waterschoot, G. Rombouts, M. Moonen, Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement, Signal Process. 88 (3) (2008) 594–611.

[12] J. F. de Andrade, Jr., M. L. R. de Campos, J. A. Apolinário, Jr., $\mathbf{L}_1$ -constrained normalized LMS algorithms for adaptive beamforming, IEEE Transactions on Signal Processing 63 (24) (2015) 6524–6539.

[13] Y. Shi, J. Zhang, K. B. Letaief, Robust group sparse beamforming for multicast green cloud-RAN with imperfect CSI, IEEE Transactions on Signal Processing 63 (17) (2015) 4647–4659.

[14] C. R. Berger, S. Zhou, J. C. Preisig, P. Willet, Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing, IEEE Transactions on Signal Processing 58 (3) (2010) 1708–1721.

[15] E. Panayirci, H. Senol, M. Uysal, H. V. Poor, Sparse channel estimation and equalization for OFDM-based underwater cooperative systems with amplify-and-forward relaying, IEEE Transactions on Signal Processing 64 (1) (2016) 214–228.

[16] D. Comminiello, S. Scardapane, M. Scarpiniti, R. Parisi, A. Uncini, Online selection of functional links for nonlinear system identification, in: S. Bassis, A. Esposito, F. C. Morabito (Eds.), Recent Advances of Neural Network Models and Applications – Proceedings of the 24th Workshop of the Italian Neural Networks Society (SIREN), May 15-16, Vietri sul Mare, Salerno, Italy, Smart Innovation, Systems and Technologies, Springer-Verlag, 2015.

[17] M. Wu, B. Schölkopf, G. Bakır, A direct method for building sparse kernel learning algorithms, The Journal of Machine Learning Research 7 (2006) 603–624.

[18] W. Liu, I. Park, J. C. Príncipe, An information theoretic approach of designing sparse kernel adaptive filters, IEEE Transactions on Neural Networks 20 (12) (2009) 1950–1961.

[19] S. Gao, I. W. H. Tsang, L. T. Chia, Sparse representation with kernels, IEEE Transactions on Image Processing 22 (2) (2013) 423–434.

[20] J. A. Bazerque, G. B. Giannakis, Nonparametric basis pursuit via sparse kernel-based learning: A unifying view with advances in blind methods, IEEE Signal Processing Magazine 30 (4) (2013) 112–125.

[21] Z. Fu, G. Lu, K. M. Ting, D. Zhang, Learning sparse kernel classifiers for multi-instance classification, IEEE Transactions on Neural Networks and Learning Systems 24 (9) (2013) 1377–1389.

[22] L. Yao, Genetic algorithm based identification of nonlinear systems by sparse Volterra filters, IEEE Transactions on Signal Processing 47 (12) (1999) 3433–3435.

[23] V. Kekatos, G. B. Giannakis, Sparse Volterra and polynomial regression models: Recoverability and estimation, IEEE Transactions on Signal Processing 59 (12) (2011) 5907–5920.

[24] D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-García, A. Uncini, Functional link adaptive filters for nonlinear acoustic echo cancellation, IEEE Transactions on Audio, Speech, and Language Processing 21 (7) (2013) 1502–1512.

[25] D. Comminiello, M. Scarpiniti, R. Parisi, A. Uncini, Convergence properties of nonlinear functional link adaptive filters, Electronics Letters 49 (14) (2013) 873–875.

[26] D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-García, A. Uncini, Nonlinear acoustic echo cancellation based on sparse functional link representations, IEEE/ACM Transactions on Audio, Speech, and Language Processing 7 (22) (2014) 1172–1183.

[27] D. Comminiello, S. Scardapane, M. Scarpiniti, R. Parisi, A. Uncini, Functional link expansions for nonlinear modeling of audio and speech signals, in: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1–8.

[28] S. Haykin, Neural Networks and Learning Machines, 3rd Edition, Prentice Hall, Upper Saddle River, NJ, 2008.

[29] J. Arenas-García, L. A. Azpicueta-Ruiz, M. T. M. Silva, V. H. Nascimento, A. H. Sayed, Combinations of adaptive filters: Performance and

convergence properties, IEEE Signal Processing Magazine 33 (1) (2016) 120–140.

[30] J. Arenas-García, A. R. Figueiras-Vidal, A. H. Sayed, Mean-square performance of a convex combination of two adaptive filters, IEEE Transactions on Signal Processing 54 (3) (2006) 1078–1090.

[31] J. Arenas-García, M. Martínez-Ramón, A. Navia-Vázquez, A. R. Figueiras-Vidal, Plant identification via adaptive combination of transversal filters, Signal Processing 86 (9) (2006) 2430–2438.

[32] J. Arenas-García, A. R. Figueiras-Vidal, Adaptive combination of proportionate filters for sparse echo cancellation, IEEE Transactions on Audio, Speech, and Language Processing 17 (6) (2009) 1087–1098.

[33] D. Comminiello, M. Scarpiniti, R. Parisi, A. Uncini, Combined adaptive beamforming schemes for nonstationary interfering noise reduction, Signal Processing 93 (12) (2013) 3306–3318.

[34] D. Comminiello, M. Scarpiniti, S. Scardapane, R. Parisi, A. Uncini, Improving nonlinear modeling capabilities of functional link adaptive filters, Neural Networks 69 (2015) 51–59.

[35] N. V. George, A. Gonzales, Convex combination of nonlinear adaptive filters for active noise control, Applied Acoustics 76 (2014) 157–161.

[36] D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-García, A. Uncini, A nonlinear architecture involving a combination of proportionate functional link adaptive filters, in: 23rd European Signal Processing Conference (EUSIPCO), Nice, France, 2015, pp. 2919–2923.

[37] L. A. Azpicueta-Ruiz, M. Lázaro-Gredilla, A. R. Figueiras-Vidal, J. Arenas-García, A block-based approach to adaptively bias the weights of adaptive filters, in: IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Beijing, China, 2011, pp. 1–6.

[38] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-García, W. Kellermann, Improved acoustic echo cancellation for low SNR based on blockwise combination of filters, in: Proceedings of the 20th International Congress on Acoustics (ICA), Sidney, Australia, 2010, pp. 1–7.

[39] A. H. Sayed, Adaptive Filters, John Wiley & Sons, Hoboken, NJ, 2008.

[40] A. Uncini, Fundamentals of Adaptive Signal Processing, Signal and Communication Technology, Springer International Publishing AG, Cham, Switzerland, 2015, ISBN 978-3-319-02806-4.

[41] J. C. Patra, R. N. Pal, B. N. Chatterji, G. Panda, Identification of nonlinear dynamic systems using functional link artificial neural networks, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 29 (2) (1999) 254–262.

[42] R. Salem, A. Zygmund, The approximation by partial sums of Fourier series, Transactions of the American Mathematical Society 59 (1) (1946) 14–22.

[43] J. Benesty, S. L. Gay, An improved PNLMS algorithm, in: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol. 2, Orlando, FL, 2002, pp. 1881–1884.

[44] H. Deng, M. Doroslovački, Improving convergence of the PNLMS algorithm for sparse impulse response identification, IEEE Signal Processing Letters 12 (3) (2005) 181–184.

[45] F. das Chagas de Souza, O. J. Tobias, D. R. Morgan, A PNLMS algorithm with individual activation factors, IEEE Transactions on Signal Processing 58 (4) (2010) 2036–3047.

[46] M. O. Sayin, Y. Yilmaz, A. Demir, S. S. Kozat, The Krylov-proportionate normalized least mean fourth approach: Formulation and performance analysis, Signal Procesing 109 (2015) 1–13.

[47] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-García, W. Kellermann, Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation, IEEE Transactions on Audio, Speech and Language Processing 19 (1) (2011) 97–110.

[48] M. Lázaro-Gredilla, L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, J. Arenas-García, Adaptively biasing the weights of adaptive filters, IEEE Transactions on Signal Processing 58 (7) (2010) 3890–3895.