Improving nonlinear modeling capabilities of functional link adaptive filters

Danilo Comminiello *, Michele Scarpiniti, Simone Scardapane, Raffaele Parisi, Aurelio Uncini

Department of Information Engineering, Electronics and Telecommunications (DIET), “Sapienza” University of Rome, Via Eudossiana 18, 00184 Rome, Italy

HIGHLIGHTS

• This paper proposes an improved split functional link adaptive filter (SFLAF).
• The proposed model is characterized by the adaptive combination of two APA filters.
• An advanced scheme is also proposed involving the combination of multiple filters.
• The adaptive combinations are performed for all the projections of the APA filters.
• The proposed models are assessed in three different nonlinear modeling problems.

ABSTRACT

The functional link adaptive filter (FLAF) represents an effective solution for online nonlinear modeling problems. In this paper, we take into account a FLAF-based architecture, which separates the adaptation of linear and nonlinear elements, and we focus on the nonlinear branch to improve the modeling performance. In particular, we propose a new model that involves an adaptive combination of filters downstream of the nonlinear expansion. Such combination leads to a cooperative behavior of the whole architecture, thus yielding a performance improvement, particularly in the presence of strong nonlinearities. An advanced architecture is also proposed involving the adaptive combination of multiple filters on the nonlinear branch. The proposed models are assessed in different nonlinear modeling problems, in which their effectiveness and capabilities are shown.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Nonlinear modeling problems have always aroused great interest in the research community. In particular, applications requiring an online modeling of nonlinearities have led to the development of many linear-in-the-parameter (LIP) nonlinear models, which consist in a nonlinear expansion of the input followed by a linear model. This approach derives from Cover’s Theorem on the separability of patterns (see Haykin, 2008), which ensures universal approximation capabilities given a sufficiently large number of nonlinear elements.

Among the family of LIP nonlinear models for online learning, representative examples include adaptive Volterra models (Azpicueta-Ruiz, Zeller, Figueiras-Vidal, Kellermann, & Arenas-García, 2013; Zhao & Zhang, 2009a), regularized networks (Poggio & Girosi, 1990; Solazzi & Uncini, 2004), spline adaptive filters (Scarpiniti, Comminiello, Parisi, & Uncini, 2013; Vecchi, Piazzelli, & Uncini, 1998), even mirror Fourier nonlinear filters (Carini & Sicuranza, 2011), kernel adaptive filters (Fan & Song, 2014; Zhu, Chen, & Príncipe, 2012), online extreme learning machines (Huang, Huang, Song, & You, 2015; Scardapane, Comminiello, Scarpiniti, & Uncini, in press). In this work, we focus on a class of LIP nonlinear filters based on the functional links (Pao, 1989; Pao & Beer, 1988). The functional link is a functional operator, which allows to represent an input pattern in a feature space where its processing turns out to be enhanced. The functional links have been widely used in single-layer feedforward neural networks, named functional link artificial neural networks (FLANNs), or also functional link artificial neural networks (FLNs) (Amin, Savitha, Amin, & Murase, 2012; Patra, Pal, Chatterji, & Panda, 1999; Scardapane, Wang, Panella, & Uncini, 2015; Zhao & Zhang, 2009b). They have also been used in conjunction with adaptive filters for online learning applications, with the name of FLANN filters (Sicuranza & Carini, 2011) or also functional link adaptive filters (FLAFs) (Comminiello, Azpicueta-Ruiz, Scarpiniti,
In this paper, we take into account a functional link-based filter, the split FLAF (SFLAF) (Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013), which separates the adaptation of linear and nonlinear elements, thus performing two distinct tasks in parallel: the estimation of the linear impulse response and the modeling of nonlinearities. With respect to prior works on FLAFs (Comminiello et al., 2011; Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013), we propose a new architecture, called combined SFLAF (CSFLAF), in which the nonlinear path is characterized by an adaptive combination of two filters downstream of the nonlinear expansion. Adaptive combination of filters exploits the diversity of parallel branches to improve the filtering performance when no much information is a priori provided on the model of signal to be processed (Arenas-García, Martínez-Ramón, Navia-Vázquez, & Figueiras-Vidal, 2006). In this regard, many efforts have been made in the linear case (Arenas-García et al., 2006; Comminiello, Scarpiniti, Parisi, & Uncini, 2013; Silva & Nascimento, 2008), but in the nonlinear case the combined output takes into account directly the joint effect of linear and nonlinear filtering (Azpicueta-Ruiz, Zeller, Figueiras-Vidal, Arenas-García, & Kellermann, 2011; Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013).

Here, we investigate the effects of the combination of two purely nonlinear outputs that leads to an improvement of the task of nonlinear modeling. The two adaptive filters on the nonlinear branch are updated by using the same affine projection algorithm (APA) (Ozeki & Umeda, 1984) but with different projection orders. In particular, choosing a unitary projection order for one filter and a higher order for the other one, we provide the two filters with different adaptation rules, respectively a gradient-based one and a Hessian-based one. This gives robustness to the model, which is able to provide improved performance, especially when an unknown system introduces very strong nonlinearities. Another novel insight in this architecture is represented by the fact that the adaptive combination is performed involving not only the current projection, as in Arévalo, Apolinário, de Campos, and Sampaio-Neto (2013), Comminiello, Scarpiniti, Parisi et al. (2013) and Ferrer, de Diego, González, and Piñero (2009), but all the available ones.

We also propose an advanced combined architecture involving the adaptive combination of three APAs downstream of the functional link expansion. This model further improves the nonlinear modeling performance by taking advantage of the capabilities of the individual filters. The proposed models are assessed in several nonlinear system identification problems showing the performance capabilities of the combined architectures that, according to the system to be identified, can just select the best performing filter or take advantage of all the filters, giving rise to an emerging learning behavior.

The rest of the paper is organized as follows: the nonlinear FLAF model is described in Section 2 and then, the proposed combined FLAF-based architecture is presented in Section 3. The advanced combined architecture is described in Section 4, while experimental results in Section 5 prove the effectiveness of the proposed architectures in different nonlinear modeling scenarios. Finally, in Section 6 our conclusions are drawn.

1.1. 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 \( x_n \in \mathbb{R}^M = [x[n] \ x[n-1] \ \cdots \ x[n-M+1]]^T \), 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 \( w_n \in \mathbb{R}^M = [w_0[n] \ w_1[n] \ \cdots \ w_{M-1}[n]]^T \), where \( w_i[n] \) is the generic \( i \)-th individual entry at the \( n \)-th time instant. The index related to a generic \( j \)-th filter is denoted as subscript, before the time index for vectors and matrices, e.g. \( w_{j,n} \). All vectors are represented as column vectors.

2. Nonlinear functional link adaptive filter

The FLAF model is based on the representation of the input signal in a higher-dimensional space (Pao, 1989), where an enhanced nonlinear modeling is allowed. Such approach derives from machine learning theory, more precisely from Cover’s Theorem on the separability of patterns (see for example Haykin, 2008).

The purely nonlinear FLAF is composed of two main parts: 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, which might be a subset of a complete set of orthonormal basis functions satisfying universal approximation constraints. The term “functional links” actually refers to the functions contained in the chosen set \( \Phi = \{ \phi_0(\cdot), \phi_1(\cdot), \ldots, \phi_{Q-1}(\cdot) \} \), where \( Q \) is the number of functional links. As depicted in Fig. 2, at the \( n \)-th time instant, the FEB receives the input sample \( x[n] \), which is stored in an input buffer \( x_n \in \mathbb{R}^{M_n} = [x[n] \ x[n-1] \ \cdots \ x[n-M_n+1]]^T \), where \( M_n \) is defined as the input buffer length. Each element of \( x_n \) is passed as argument to the chosen set of functions \( \Phi \), thus yielding a subvector \( \bar{g}_{i,n} \in \mathbb{R}^Q \),

\[
\bar{g}_{i,n} = \begin{bmatrix}
\phi_0(x[n-i]) \\
\phi_1(x[n-i]) \\
\vdots \\
\phi_{Q-1}(x[n-i])
\end{bmatrix}.
\]
The concatenation of all the subvectors, for $i = 0, \ldots, M_{in} - 1$, engenders an expanded buffer $\mathbf{g}_n \in \mathbb{R}^{M_{en}}$:

$$
\mathbf{g}_n = [\mathbf{g}_{0,n}^T \mathbf{g}_{1,n}^T \cdots \mathbf{g}_{M_{in}-1,n}^T]^T
$$

(2)

where $M_{en} \geq M_{in}$ represents the length of the expanded buffer. Note that $M_{en} = M_{in}$ only when $Q = 1$. As functional expansion, we choose a nonlinear trigonometric series expansion:

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

(3)

where $p = 1, \ldots, P$ is the expansion index, $P$ being the expansion order, and $j = 0, \ldots, Q - 1$ is the functional link index. The trigonometric expansion implies a functional link set $\Phi$ composed of $Q = 2P$ functional links. Some convergence properties of the nonlinear FLAF of Fig. 1 can be found in Comminiello, Scarpiniti, Parisi, and Uncini (2013b). It is worth noting that (3) actually refers to a memoryless expansion, since it does not involve cross-products, but it can be easily extended to a memory expansion. A way of considering the memory of a nonlinearity is taking into account the outer products of the input sample with the functional links of the previous input samples (see Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013 for a detailed explanation).

The expanded buffer $\mathbf{g}_n$ is then fed into a linear adaptive filter $\mathbf{w}_{FL,n} \in \mathbb{R}^{M_{en}} = [\mathbf{w}_{FL,0}^T \mathbf{w}_{FL,1}^T \cdots \mathbf{w}_{FL,M_{en}-1}^T]^T$, thus providing the nonlinear output:

$$
y_{FL}[n] = \mathbf{g}_n^T \mathbf{w}_{FL,n-1}
$$

(4)

Thereby, the nonlinear error signal is:

$$
e_{FL}[n] = d[n] - y_{FL}[n]
$$

(5)

which is used for the adaptation of $\mathbf{w}_{FL,n}$. In (5), $d[n]$ represents the desired signal for the nonlinear model. Being $\mathbf{w}_{FL,n}$ a conventional linear filter, it can be adapted by any adaptive algorithm based on the minimization of the mean square error (see for example Sayed, 2008 and Uncini, 2015). The use of an adaptive filter after the nonlinear expansion allows to apply the FLAF model to several online learning applications.

3. The combined split functional link adaptive filter

Nonlinear FLAFs are used to build up adaptive filtering architectures for nonlinear modeling (Comminiello et al., 2011; Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013; Comminiello, Scarpiniti, Azpicueta-Ruiz, Arenas-García, & Uncini, 2014). The simplest one is the split FLAF (SFLAF), which is composed of a nonlinear FLAF in parallel with a linear adaptive filter (Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013). This allows to consider two different choices about the settings of the two filtering branches, thus optimizing the linear and the nonlinear modeling. In this work, we propose a variant of the SFLAF, in which we exploit the combination of adaptive filters to improve the nonlinear modeling performance. This architecture, denoted as combined SFLAF (cSFLAF), is depicted in Fig. 3, where it is possible to notice that, similarly to the SFLAF in Comminiello, Scarpiniti, Azpicueta-Ruiz et al. (2013), 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 a Hessian-based and a gradient-based algorithm. In order to generalize the description we use the APA to adapt all the linear filters of the structure.

The input signal $x[n]$ is collected in a data matrix $\mathbf{X}_n \in \mathbb{R}^{K_L \times M}$ for the linear branch:

$$
\mathbf{X}_n = \begin{bmatrix}
\mathbf{x}_n & \mathbf{x}_{n-1} & \cdots & \mathbf{x}_{n-K_L+1}
\end{bmatrix}^T
$$

(6)

where $K_L$ is the memory of entries at previous time instants, i.e., the projection order of the input data matrix. Data are processed by the linear adaptive filter $\mathbf{w}_{L,n}$, thus yielding the linear output contribution $\mathbf{y}_{L,n} \in \mathbb{R}^{K_L}$:

$$
\mathbf{y}_{L,n} = \mathbf{X}_n \mathbf{w}_{L,n-1}
$$

(7)

The linear adaptive filter $\mathbf{w}_{L,n}$ aims at estimating the unknown linear impulse response. Its update equation using the APA is:

$$
\mathbf{w}_{L,n} = \mathbf{w}_{L,n-1} + \mu_L \mathbf{X}_n^T (\delta \mathbf{I} + \mathbf{X}_n \mathbf{X}_n^T)^{-1} \mathbf{e}_{L,n}
$$

(8)
where $\mu_k$ and $\delta_k$ are respectively the step-size parameter and the regularization factor for the linear path. In (8), the error buffer $e_{n} \in \mathbb{R}^{K_1} = [e_{n}, e_{n-1}, \ldots, e_{n-K_1+1}]^T$ is a selection of the overall error signal containing the first $K_1$ samples.

We adopt the APA to update also the coefficients of the filters on the nonlinear branch, $w_{i,j}$ for $j = \{FL1, FL2\}$, each one characterized by its own projection order $K_j$. Therefore, the input data matrix processed by the linear branch, or a selection of it (see Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013), is fed at the same time into the FEB on the nonlinear branch, as explained in the previous section, thus yielding an expanded buffer matrix, whose selection depends on the projection orders, i.e.:

$$G_{n,j} \in \mathbb{R}^{K_j \times Mn} = [g_{n}, g_{n-1}, \ldots, g_{n-K_j+1}]^T. \quad (9)$$

Therefore, the individual outputs $y_{i,j} \in \mathbb{R}^{K_j}$ of the two filters on the nonlinear path are:

$$y_{i,j} = G_{n,j}w_{i,j-1}. \quad (10)$$

We define now the desired signal buffers $d_{i} \in \mathbb{R}^{K_i}$ and $d_{i,j}$, for $j = \{FL1, FL2\}$, which contain the previous samples of the desired signal $d[n]$ according to the corresponding projection order. Hence, it is possible to define the following error signal vectors:

$$e_{n,j} \in \mathbb{R}^{K_j} = \begin{cases} d_{n} - y_{i,j,n} - [y_{i,j,n}; 0], & \text{for } K_i \geq K_j \\ d_{n,j} - [y_{i,j,n}; 0] - y_{i,j,n}, & \text{for } K_i < K_j \end{cases}. \quad (11)$$

where $0 \in \mathbb{R}^{K_i-K_j}$ is a vector of zeros used for padding. Error signals in (11) are used to adapt the filters $w_{i,j} \in \mathbb{R}^{K_i}$, for $j = \{FL1, FL2\}$:

$$w_{i,j} = w_{i,j,n-1} + \mu_t G_{n,j}^T (\delta_{1} + 2G_{n,j}^T G_{n,j})^{-1} e_{n,j}. \quad (12)$$

It is worth noting that the error signals that adapt the filters on the nonlinear path also involve the linear output contribution, as in the SFLAF (Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013).

As it is possible to see in Fig. 3, the overall output of the nonlinear path is achieved by combining convexly the individual filter outputs (4):

$$y_{N,n} = \begin{cases} \lambda_n \odot y_{FL1,n} + \left(1 - \lambda_n\right) \odot \left[y_{FL2,n}; 0\right], & \text{for } K_{FL1} \geq K_{FL2} \\ \lambda_n \odot \left[y_{FL1,n}; 0\right] + \left(1 - \lambda_n\right) \odot y_{FL2,n}, & \text{for } K_{FL1} < K_{FL2} \end{cases}. \quad (13)$$

where $\lambda_n \in \mathbb{R}^{K_n}$, is the mixing parameter vector with $K_n = \max(K_{FL1}, K_{FL2})$, $\mathbf{0} \in \mathbb{R}^{K_n-K_i}$, for $j = \{FL1, FL2\}$, are vectors of zeros, and $1 \in \mathbb{R}^{K_n}$ is a vector of ones. The operator $\odot$ denotes the Hadamard product.

The elements of the mixing parameter vector are adaptive weights that balance the combination between the two filters $w_{i,j}$ ($j = \{FL1, FL2\}$) along their projections, giving more importance to the best performing filter. Such awareness is obtained according to a mean square error minimization. In particular, the adaptation of $\lambda_n$ is performed by using an auxiliary vector $a_n \in \mathbb{R}^{K_n}$, whose elements are related to those of $\lambda_n$ by means of a sigmoidal function that keeps the mixing parameter in the range $0, 1$ and defined according to Comminiello, Scarpiniti, Parisi et al. (2013) and Lazaro-Gredilla, Azpicueta-Ruiz, Figueiras-Vidal, and Arenas-Garcia (2010) as:

$$\lambda_k [n] = \beta \left(\frac{1}{1 + e^{-\alpha\lambda_k [n]}} - \alpha\right), \quad k = 0, \ldots, K_n - 1 \quad (14)$$

$$\alpha = \frac{1}{1 + e^{\delta}}, \quad \beta = \frac{1}{1 - 2\alpha}. \quad (15)$$

The auxiliary vectors are updated by using a gradient descent rule, therefore, for $k = 0, \ldots, K_n - 1$:

$$a_k [n] = a_k [n-1] + \frac{\mu_t}{\beta R_k [n-1]} e_{n,k} [n] \Delta y_k [n] \times (\lambda_k [n] + \alpha \beta) (\beta - \alpha \beta - \lambda_k [n]) \quad (16)$$

where $\Delta y_k [n] = (y_{FL1,k} [n] - y_{FL2,k} [n])$. Also, in (16), $\mu_t$ is the step-size parameter of the adaptive combination, $R_k [n] = \gamma R_k [n-1] + \left(1 - \gamma\right) \Delta y_k^2 [n]$ is the estimated power of $\Delta y_k [n]$, and $\gamma$ is a smoothing factor. The error buffer $e_{n} \in \mathbb{R}^{K_{max}}$ in (16), is composed of the first $K_1$ samples of the overall error signal, similarly to $e_{i,n}$ in (8). The overall error signal $e_{n} \in \mathbb{R}^{K_{max}}$, with $K_{max} = \max(K_{FL1}, K_{FL2})$, is derived as:

$$e_{n} = d_{n} - y_{i,n} - y_{N,n}. \quad (17)$$

where all the vectors take into account the larger projection order.

Convergence properties of the convex combination of two adaptive filters with different updating rules can be found in Comminiello, Scarpiniti, Parisi et al. (2013) and Silva and Nascimento (2008).

It is worth noting that the cSFLAF can be extended further by taking into account the combination of adaptive filters also on the linear branch, thus improving the linear modeling performance too.

4. An advanced cSFLAF involving the combination of multiple filters

A straightforward development of the cSFLAF described in the previous section involves the adaptive combination of more than two filters downstream of the FEB. For our purposes, we consider the combination of three APA filters, but it can be easily extended to a larger number of filters. Several techniques were proposed in the literature to combine multiple filters including affine and convex combination, and softmax activation function (Arenas-Garcia, Gómez-Verdejo, & Figueiras-Vidal, 2005; Arévalo et al., 2013; Azpicueta-Ruiz, Zeller, Figueiras-Vidal, & Arenas-Garcia, 2010; Kozat, Erdogan, Singer, & Sayed, 2010). Among these methods, we follow the scheme proposed in Arévalo et al. (2013) and we include it in the cSFLAF, thus resulting in an advanced scheme, depicted in Fig. 4, that we denote as c3SFLAF.

Each filter on the nonlinear path is adapted by using an APA with different projection order. In particular, we set the smallest order $K_{FL1}$ for the first filter, a higher order for the second filter $K_{FL2}$, and the highest for the third filter, i.e. $K_{FL3} \leq K_{FL2}$. In order to make the combination stable we choose a convex combination such that the output of the nonlinear path can be written as:

$$y_{N,n} = \eta_{1,n} \odot \left[y_{FL1,n}; 0_{FL1}\right] + \eta_{2,n} \odot \left[y_{FL2,n}; 0_{FL2}\right] + \eta_{3,n} \odot \left[y_{FL3,n}; 0_{FL3}\right] \quad (18)$$

where $\eta_{i,n} \in \mathbb{R}^{K_n}$ are the mixing parameter vectors, for $i = 1, 2, 3$, taking into account that in this case $K_n = K_{FL3}$ since it is the maximum order. Moreover, in (18), $0_{i} \in \mathbb{R}^{K_{FL3}-K_i}$, for $i = \{FL1, FL2\}$, are vectors of zeros. The mixing parameters are subjected to convex constraints, therefore $\eta_{i,k} [n] \in [0, 1]$ and $\sum_{i=1}^{3} \eta_{i,k} = 1$, for $i = 1, 2, 3$ and $k = 1, \ldots, K_n$. The convex combination is performed by following the two-stage procedure described in Arévalo et al. (2013), i.e. by combining the first two filters and then combining the result with the third filter. Therefore, the mixing parameters in (18) can be expressed in terms of the mixing parameters used in the two stages:

$$\eta_{1,n} = \lambda_{1,n} \odot \lambda_{2,n} \quad (19)$$

$$\eta_{2,n} = (1 - \lambda_{1,n}) \odot \lambda_{2,n} \quad (20)$$

$$\eta_{3,n} = 1 - \lambda_{2,n} \quad (21)$$
where $\lambda_{l,n} \in [0, 1]$, with $l = 1, 2$, and $1 \in \mathbb{R}^{K_N}$ is a vector of ones.

The adaptation of the mixing parameters $\lambda_{l,n}$, for $l = 1, 2$, is computed similarly to (14) as:

$$
\lambda_{l,k}[n] = \beta \left( \frac{1}{1 + e^{-a_k[n]}} - \alpha \right), \quad k = 0, \ldots, K_N - 1
$$

(22)

where $\alpha$ and $\beta$ are the same of (15). The auxiliary vectors $a_k[n]$, for $l = 1, 2$ and $k = 0, \ldots, K_N - 1$, are still updated by minimizing the squared error, i.e.

$$
a_k[n] = a_k[n - 1] + \frac{\mu_e}{2} \frac{\partial (e_{N,k}[n])^2}{\partial a_k[n - 1]}.
$$

(23)

In this case the error $e_{N,k}[n]$ is provided by the combination of the three local errors, as done for the output in (18), and it can be expressed for $k = 0, \ldots, K_N - 1$ as:

$$
e_{N,k}[n] = \lambda_{1,k}[n] \lambda_{2,k}[n] e_{FL1,k}[n] + (1 - \lambda_{1,k}[n]) \lambda_{2,k}[n] e_{FL2,k}[n] + (1 - \lambda_{2,k}[n]) e_{FL3,k}[n].
$$

(24)

Solving the derivatives of the squared error in (23) for $l = 1$, we obtain:

$$
\frac{\partial (e_{N,k}[n])^2}{\partial a_{1,k}[n - 1]} = \frac{\partial (e_{N,k}[n])^2}{\partial \lambda_{1,k}[n - 1]} \cdot \frac{\partial \lambda_{1,k}[n]}{\partial a_{1,k}[n - 1]}
$$

(25)

where

$$
\frac{\partial (e_{N,k}[n])^2}{\partial \lambda_{1,k}[n - 1]} = -2e_{N,k}[n] (y_{FL1,k}[n] - y_{FL2,k}[n]) \lambda_{2,k}[n]
$$

(26)

and

$$
\frac{\partial \lambda_{1,k}[n]}{\partial a_{1,k}[n - 1]} = \beta e^{-a_{1,k}[n]} \left( 1 - e^{-a_{1,k}[n]} \right)^2
$$

$$
= \left( \lambda_{1,k}[n] + \alpha \beta \right) \frac{1}{\beta} (\beta - \alpha \beta - \lambda_{1,k}[n]).
$$

(27)

Therefore, taking into account (24), (26), (27), the auxiliary parameter (23) for $l = 1$ can be written as:

$$
a_{1,k}[n] = a_k[n - 1] + \frac{\mu_e}{\beta r_{1,k}[n - 1]} e_{N,k}[n] \left[ (y_{FL1,k}[n] - y_{FL2,k}[n]) \lambda_{2,k}[n] \right.\
$$

$$
- \left. y_{FL2,k}[n] \lambda_{2,k}[n] \cdot (\lambda_{1,k}[n] + \alpha \beta) \times (\beta - \alpha \beta - \lambda_{1,k}[n]) \right].
$$

(28)

where the normalization term $r_{1,k}[n]$ is derived similarly to Azpicueta-Ruiz, Figueiras-Vidal, and Arenas-García (2008) as:

$$
r_{1,k}[n] = y_{r_{1,k}[n - 1]} + (1 - \gamma) \left[ y_{FL1,k}[n] - y_{FL2,k}[n] \right].
$$

(29)

Similarly, we solve the derivatives in (23) for $l = 2$, thus achieving:

$$
\frac{\partial (e_{N,k}[n])^2}{\partial a_{2,k}[n - 1]} = \frac{\partial (e_{N,k}[n])^2}{\partial \lambda_{2,k}[n - 1]} \cdot \frac{\partial \lambda_{2,k}[n]}{\partial a_{2,k}[n - 1]}
$$

(30)

where

$$
\frac{\partial (e_{N,k}[n])^2}{\partial \lambda_{2,k}[n - 1]} = -2e_{N,k}[n] (y_{FL1,k}[n] - y_{FL2,k}[n])
$$

$$
+ y_{FL1,k}[n] - y_{FL3,k}[n]
$$

(31)

and

$$
\frac{\partial \lambda_{2,k}[n]}{\partial a_{2,k}[n - 1]} = \frac{\beta e^{-a_{2,k}[n]}}{\left( 1 - e^{-a_{2,k}[n]} \right)^2}
$$

$$
= \left( \lambda_{2,k}[n] + \alpha \beta \right) \frac{1}{\beta} (\beta - \alpha \beta - \lambda_{2,k}[n]).
$$

(32)

Hence, considering (24), (31), (32), the auxiliary parameter (23) for $l = 2$ can be written as:

$$
a_{2,k}[n] = a_k[n - 1] + \frac{\mu_e}{\beta r_{2,k}[n - 1]} e_{N,k}[n] \left[ (y_{FL1,k}[n] - y_{FL2,k}[n]) \lambda_{2,k}[n] \right.\
$$

$$
- \left. y_{FL2,k}[n] \lambda_{2,k}[n] \cdot (\lambda_{1,k}[n] + \alpha \beta) \times (\beta - \alpha \beta - \lambda_{2,k}[n]) \right].
$$

(33)
where
\[ r_{2,k}[n] = γ r_{2,k}[n-1] + (1-γ) \left[ y_{FL1,k}[n] y_{FL2,k}[n] \right. \\
- \left. y_{FL2,k}[n] + y_{FL2,k}[n] - y_{FL3,k}[n] \right]^2. \] (34)

The overall error signal of the c3SFLAF scheme is then computed similarly to (17).

It is worth noting that, in order to compute efficiently the matrix inversions of each APA filter on the nonlinear path, a recursive algorithm based on Schur's complement (Arévalo et al., 2013) can be used, which takes advantage of the computation of the inverse matrix with increasing order.

5. Experimental results

In this section, we evaluate the performance improvement of the proposed combined FLAF-based architectures with respect to the corresponding individual SFLAFs. We assess the proposed architecture in different scenarios to provide a comprehensive view of it.

5.1. Nonlinear system identification scenario #1

In this set of experiments, we evaluate the combined SFLAF architectures in a scenario involving a very strong nonlinearity, which is composed of a cascade of linear–nonlinear–linear blocks, as depicted in Fig. 5. In particular, the first block is a 4-th order IIR digital Butterworth filter, whose transfer function is (Panicker, Mathews, & Sicuranza, 1998):
\[ H_{\lambda}(z) = \frac{(0.2851 + 0.5704z^{-1} + 0.2851z^{-2})}{(1 - 0.1024z^{-1} + 0.4475z^{-2})} \cdot \frac{(0.2851 + 0.5701z^{-1} + 0.2851z^{-2})}{(1 - 0.0736z^{-1} + 0.0408z^{-2})}. \] (35)

The resulting output \( y_{\lambda}[n] \) is then fed into a nonlinear system, which applies the following sigmoid function:
\[ y_{\beta}[n] = \xi \left( \frac{1}{1 + e^{-\rho y_{\lambda}[n]}} - \frac{1}{2} \right) \] (36)
where the parameter \( \xi \) is a sigmoid gain and it is set equal to \( \xi = 2 \), while \( \rho \) represents the sigmoid slope and it is chosen as:
\[ \rho = \begin{cases} 4, & y_{\lambda}[n] > 0 \\ \frac{1}{2}, & y_{\lambda}[n] \leq 0 \end{cases} \] (37)
The signal \( y_{\beta}[n] \) is then processed by a 4-th order IIR Chebyshev filter with transfer function (Panicker et al., 1998):
\[ H_{\beta}(z) = \frac{(0.2025 + 0.2880z^{-1} + 0.2025z^{-2})}{(1 - 1.01z^{-1} + 0.5861z^{-2})} \cdot \frac{(0.2025 + 0.0034z^{-1} + 0.2025z^{-2})}{(1 - 0.6591z^{-1} + 0.1498z^{-2})}. \] (38)
thus yielding the output \( y_{\beta}[n] \). Finally, the desired signal can be obtained just by adding Gaussian noise \( v[n] \) with variance \( \sigma_v^2 = 0.01 \):
\[ d[n] = y_{\beta}[n] + v[n]. \] (39)
The input signal to the described system is generated by a first-order autoregressive model, whose transfer function is \( \sqrt{1-\theta^2} / (1-\theta z^{-1}) \), with \( \theta = 0.8 \), fed with an i.i.d. Gaussian random process. The length of the input signal is \( L = 10,000 \) samples.

In order to model the described system, we use a cSFLAF characterized by a linear filter with \( K_1 = 2 \), and the combination of two APA filters on the nonlinear path, having respectively \( K_{FL1} = 1 \) and \( K_{FL2} = 3 \). Other parameters relating to the filters are chosen also according to the studies in Comminiello et al. (2011); Comminiello, Scarpiniti, Azpicueta-Ruiz et al. (2013); Comminiello et al. (2013b); \( \mu_1 = 0.2, \mu_{FL1} = \mu_{FL2} = 0.1, \delta = 10^{-3} \) (same for all the filters), \( M = 7 \). Concerning the setting of the FEB, we choose a memory-expansion with an expansion order of \( P = 30 \) and \( M_m = M \), while the adaptive combination is characterized by: \( \mu_c = 0.5, a_0[n] = 0 \) and \( r_1[n] = 1 \) for \( k = 0, \ldots, K_n - 1 \). We compare the cSFLAF with the individual SFLAFs with \( K_{FL1} \) and \( K_{FL2} \). It is worth noting that the SFLAF with \( K_{FL1} \) is nothing but the architecture proposed and assessed in Comminiello et al. (2011); Comminiello, Scarpiniti, Azpicueta-Ruiz et al. (2013). Since our aim in this work is to focus on the performance improvement brought by the combined architectures over the SFLAF, we do not present comparisons with other approaches, which can be found in the aforementioned works (Comminiello et al., 2011; Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013).

Performance is evaluated in terms of the excess mean square error (EMSE), in dB:
\[ \text{EMSE}[n] = 10 \log_{10} \left( E \left[ (e[n] - v[n])^2 \right] \right) \] (40)
which is averaged over \( 10,000 \) runs with respect to input and noise. Results in Fig. 6 show that initially the cSFLAF exploits the faster speed of convergence of the SFLAF with \( K_{FL1} \), then it follows the SFLAF with \( K_{FL2} \) to boost the convergence rate, while at steady state it behaves again like the SFLAF with \( K_{FL1} \), which achieves the best performance.

We repeat the experiment considering a c3SFLAF architecture, in which we add a third APA filter on the nonlinear branch with \( K_{FL3} = 6 \). We compare the performance of this architecture with the three individual SFLAFs. Results are depicted in Fig. 7(a), where it is possible to notice that the combined architecture exploits the three filters on the nonlinear branch, thus always yielding the best performance. It is worth noting that, although the SFLAF with \( K_{FL3} \) never achieves the best performance with respect to the other two SFLAFs, it gives its active contribution to improve the performance.
of the c3SFLAF architecture. In fact, comparing the performance of the cSFLAF in Fig. 6 and the c3SFLAF in Fig. 7(a), it is possible to notice a further gain for the c3SFLAF during the convergence. The contribution of each APA filter of the nonlinear branch in the c3SFLAF is highlighted in Fig. 7(b), where it is possible to notice the evolution of the three mixing parameters during the whole length of the experiment.

5.2. Nonlinear system identification scenario #2

We consider another scenario involving a parallel scheme, where one branch is purely linear, while another branch is composed of a cascade of linear–nonlinear–linear blocks, as depicted in Fig. 8. The first branch, composed of an FIR filter formed with $M = 10$ independent random values between −1 and 1, whose convolution with the input signal generates the first branch output $\tilde{y}_1$. On the other hand, the first block on the parallel branch is a 2-nd order lowpass digital IIR Butterworth filter with normalized cutoff frequency $\omega_{nc} = 0.2$, whose transfer function is:

$$H_A(z) = \frac{0.0675 + 0.1349z^{-1} + 0.0.0675z^{-2}}{1 - 1.1430z^{-1} + 0.4128z^{-2}}.$$ (41)

The resulting output $\tilde{y}_{2A}[n]$ is then added to the output of the first branch, together with additive Gaussian noise with variance $\sigma_v^2 = 0.01$. Therefore, the desired signal is:

$$d[n] = \tilde{y}_1[n] + \tilde{y}_{2C}[n] + v[n].$$ (44)

The input signal is colored Gaussian noise, generated as in the previous experiment, with length $L = 5000$ samples. Again, we model the described system by using a cSFLAF composed of a linear filter with $K_L = 2$, and a combination of two APA filters on the nonlinear path, having respectively $K_{FL1} = 1$ and $K_{FL2} = 3$. The parameter setting is the same of the previous experiment, except of $M = 10$ and $\mu_L = 10^2$.

Results in terms of the EMSE are depicted in Fig. 9, where it is possible to see that in this case the SFLAF does not take advantage of a projection order greater than 1, since the SFLAF with $K_{FL1}$ always provides better performance than the other SFLAF. However, the cSFLAF does not restrict itself to pursue the best performing filter, but it exploits the adaptive combination thus yielding a performance improvement. With respect to the scenario #1, in this case the collaboration between the two filters on the nonlinear branch produces an emerging behavior.

Again, we assess the performance of the cSFLAF by adding a third APA on the nonlinear branch with $K_{FL3} = 6$. In Fig. 10(a), results show that even in this case the SFLAF with $K_{FL3}$ never provides better performance than the other two SFLAFs. However, its contribution improves the convergence performance of the c3SFLAF architecture. The values assumed by the mixing parameters of the three APAs on the nonlinear branch of the c3SFLAF is shown in Fig. 10(b), where it is possible to notice that the filters give a constant, but always active, contribution to the nonlinear modeling, thus confirming the superior performance provided by the combined architecture with respect to individual SFLAFs.

5.3. Nonlinear acoustic echo cancellation scenario

In the last set of experiments, we evaluate the proposed cSFLAF in a nonlinear acoustic echo cancellation (NAEC) scenario,
characterized by a simulated environment with reverberation time of $T_{60} \approx 120$ ms. The far-end signal, reproduced by a loudspeaker and captured by a microphone, is a colored signal generated as in the previous experiment. In order to introduce a nonlinearity in the echo path, which can simulate a memoryless loudspeaker distortion, we apply a symmetrical soft-clipping to the far-end signal, described by the following expression (Comminiello et al., 2014; Zölzer & Amatriain, 2002):

$$y[n] = \begin{cases} 
2x[n] & \text{for } 0 \leq |x[n]| \leq \zeta \\
\frac{3}{2} - \frac{(2 - |x[n]|)^2}{3} & \text{for } \zeta \leq |x[n]| \leq 2\zeta \\
\text{sign}(x[n]) & \text{for } 2\zeta \leq |x[n]| \leq 1
\end{cases}$$

(45)

where $\zeta$ is a nonlinearity threshold chosen as $\zeta = 0.1$. The nonlinear signal $y[n]$ is convolved with a simulated acoustic impulse response, truncated after $M = 300$ samples, using a $8\,\text{kHz}$ sampling rate. White Gaussian noise is added as background noise $v[n]$, providing $20\,\text{dB}$ of signal to noise ratio (SNR). The length of the experiment is $L = 80000$ samples, i.e., $10\,\text{s}$. In order to introduce a change in the environment, we increase the nonlinearity level by setting the threshold equal to $\zeta = 0.05$ at time instant $n = L/2$. The performance measure is the echo return loss enhancement (ERLE), defined as:

$$\text{ERLE}[n] = 10 \log_{10} \left( \frac{E\left[ |d[n]|^2 \right]}{E\left[ |e[n]|^2 \right]} \right)$$

(46)

which is estimated by time averaging.

First, we consider the performance achieved by a cSFLAF, characterized by the following parameter setting: $K_{c} = 2, K_{FL1} = 1, K_{FL2} = 2, \mu_e = 0.5$. The rest of the parameters is set as in the previous experiments. Results are shown in Fig. 11, in which it can be noticed that in both the halves of the experiment the SFLAF with $K_{FL2}$ shows better convergence performance with respect to the SFLAF with $K_{FL1}$. Both the SFLAFs show similar performance at steady state. However, the cSFLAF takes advantage of the adaptive combination of the SFLAFs, thus providing faster convergence than the individual SFLAFs.

We also assess the performance of the c3SFLAF, by adding a third APA on the nonlinear branch with $K_{FL3} = 3$. Results in terms of the ERLE are depicted in Fig. 12(a), where it is clear that the c3SFLAF exploits the convergence performance of the SFLAF with $K_{FL3}$ first, and then the contributions achieved by the other two SFLAFs at steady state. The evolution of the mixing parameters of the c3SFLAF is shown in Fig. 12(b), where it is evident that at the convergence state the c3SFLAF exploits the capabilities of the SFLAF with $K_{FL3}$, while at steady state the c3SFLAF privileges the SFLAF with $K_{FL1}$. It is also possible to notice that the SFLAF with $K_{FL2}$ is exploited in the middle part of the first half of the experiment, while in the second half, where the nonlinearity is stronger, the SFLAF with $K_{FL2}$ does not provide useful performance.

6. Conclusion

In this paper, we have proposed a combined FLAF-based architecture, which is characterized by an adaptive combination of
filters on the nonlinear branch, downstream of the functional expansion. An advanced scheme is also proposed, in which the nonlinear branch involves the adaptive combination of multiple filters. The adaptive combination allows the proposed schemes to improve their performance in the presence of strong and time-varying nonlinearities. In particular, compared to individual SFLAFs, the combined architectures pursue the behavior of the best performing SFLAF when the performance difference between the SFLAFs is significant. On the other hand, when individual SFLAFs provide similar performance, the combined models take advantage of the adaptive combination, thus yielding a performance improvement. Overall, the combined architectures always show the best behavior. Future research lines may involve the use of the adaptive combination also on the linear path, the extension of the cSFLAF also to collaborative FLAF-based architectures (Comminiello, Scarpiniti, Azpicueta-Ruiz et al., 2013), and, moreover, the development of faster constrained algorithms to perform the combination of filters on the nonlinear path.

Acknowledgment

The work of Danilo Comminiello has been partly funded by bSound.

References


