Convex Combination of Spline Adaptive Filters

Michele Scarpiniti, Danilo Comminiello, Aurelio Uncini
Department of Information Engineering, Electronics and Telecommunications (DIET),
Sapienza University of Rome
via Eudossiana 18, 00185 Rome, Italy
{michele.scarpiniti, danilo.comminiello, aurelio.uncini}@uniroma1.it

***Abstract*—In this paper, we propose an adaptive and convex combination of a recent class of nonlinear adaptive filters in different configurations. The proposed architecture relies on the properties of the adaptive combination of filters which exploits the capabilities of different constituents, thus adaptively providing at least the behavior of the best performing filter. The nonlinear functions involved in the adaptation process are based on spline function interpolation and their shapes can be modified during learning using gradient-based techniques. In addition, we derive a simple form of the adaptation algorithm and present some experimental results that demonstrate the effectiveness of the proposed method.**

***Index Terms*—Nonlinear adaptive filter, Convex combination, Flexible spline function, Constrained Least Mean Square, System identification.**

I. Introduction

ONE of the main issues in nonlinear filtering is the correct choice of the model to be used in the identification process [1]. Although many general solutions exist, such as Neural Networks (NNs) [2], Kernel Adaptive Filters (KAFs) [3], polynomial adaptive filters and Volterra Adaptive Filters (VAFs) [4], which provide good results in many simple applications, the computational complexity of such methods often prevents their use in real-world applications.

Among the several options proposed in literature, in nonlinear filtering one of the most used structures is the so-called block-oriented representation, in which dynamic linear time invariant (LTI) models are connected with memoryless nonlinear functions. The basic classes of block-oriented nonlinear systems are represented by the Wiener (cascade of a linear LTI filter followed by a static nonlinear function) and the Hammerstein (cascade of a static nonlinear function followed by a LTI filter) models [5] and by those system architectures originated by the connection of these two classes according to different topologies (i.e. parallel, cascade, feedback etc. [6]), such as Sandwich models introduced in [5], [7], composed by the connection of linear-nonlinear-linear (LNL) blocks or nonlinear-linear-nonlinear (NLN) blocks.

Recently, a novel approach based on Spline Adaptive Filters (SAFs) has been proposed in order to implement block-oriented models such as the Wiener SAF (WSAF) [8], the Hammerstein SAF (HSAF) [9] and Sandwich SAF (namely, S1SAF and S2SAF) [10]. In a general nonlinear problem, and with no *a priori* information, the adaptive model (e.g., Hammerstein, Wiener or Sandwich) which works better is the one that mostly resembles the system to be identified. In this regard, a block-oriented architecture could not work well for the identification of a general unknown nonlinear system. The idea to overcome this nonfunctional behavior is the usage of an architecture combining more nonlinear systems in a convex manner.

To this purpose, in the context of linear adaptive filtering, several architectures combining different learning algorithms were recently proposed [11]–[16]. In these works, it was proved that the convex combination of adaptive filters can provide an interesting way to improve adaptive filter performance. Such combination scheme was also applied to a particular class of nonlinear adaptive filter approach (see for example [17]–[19] or Chapter 11 in [1]).

In this paper, we extend the idea of convex filter combination to SAFs. A particular attention is posed to the derivation of the learning algorithm for the final combination by solving a suitable constrained optimization problem. In fact, the adaptation of such coefficients should be fast and accurate in order to guarantee a good convergence performance.

The rest of the paper is organized as follows. Section II introduces briefly the SAFs, while Section III describes the main idea of the paper. Then Section IV shows some experimental results, and finally Section V concludes the work.

II. Background on SAF

For a complete introduction on spline adaptive filtering and spline interpolation, we refer to our recent papers [8]–[10]. In summary, a SAF consists in the cascade of one or two LTI filters and one or two static nonlinear function, depending on which model among the Wiener, Hammerstein or Sandwich ones has to be implemented. The peculiarity of SAFs is that the static nonlinearity is obtained as a cubic spline interpolation of a fixed number of points collected in a look-up table (LUT) and called as *control points*. By adapting the values of the control points into the LUT, it is possible to change the shape of the interpolated nonlinearity. Specifically, the output $\beta[n]$ of the nonlinear function $\beta[n] = \varphi(\alpha[n])$ is determined by using two local parameters: u_n and i, which directly depend on the input $\alpha[n]$. In the simple case of a uniform spacing

This work has been supported by the project: "GAUChO — A Green Adaptive Fog Computing and networking Architectures" funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 – grant 2015YPXH4W_004, and by the projects: "Vehicular Fog energy-efficient QoS mining and dissemination of multimedia Big Data streams (V-FoG and V-Fog2)" and "SoFT: Fog of Social IoT", funded by Sapienza University of Rome Bando 2016, 2017 and 2018.

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

of knots and a third-order curve interpolation adopted in this work, the computation procedure for the determination of the span index i and the local parameters u_n can be expressed by the following equations [8]:

$$u_n = \frac{\alpha[n]}{\Delta x} - \left\lfloor \frac{\alpha[n]}{\Delta x} \right\rfloor, \qquad i = \left\lfloor \frac{\alpha[n]}{\Delta x} \right\rfloor + \frac{Q-1}{2}, \tag{1}$$

where Δx is the uniform space between knots, $\lfloor \bullet \rfloor$ is the floor operator and Q is the total number of control points. Note that the index i is depending of time n, i.e. i_n; for simplicity of notation we adopt the convention $i_n \equiv i$.

The output of the nonlinearity can be then evaluated as follows:

$$\beta[n] = \varphi\left(\alpha[n]\right) = \varphi_i(u_n) = \mathbf{u}_n^T \mathbf{C} \mathbf{q}_{i,n}, \tag{2}$$

where, considering a third-order spline basis, the matrix $\mathbf{C} \in \mathbb{R}^{4\times 4}$ is a pre-computed matrix, usually called *spline basis matrix*, the vector $\mathbf{u}_n$ is defined as $\mathbf{u}_n \in \mathbb{R}^{4\times 1} = \left[u_n^3,\, u_n^2,\, u_n,\, 1\right]^T$, and the vector $\mathbf{q}_{i,n}$ contains the control points at instant n and is defined by $\mathbf{q}_{i,n} \in \mathbb{R}^{4\times 1} = \left[q_i,\, q_{i+1},\, q_{i+2},\, q_{i+3}\right]^T$, where q_k is the k-th entry in the LUT. In (2), $\varphi_i(u_n)$ is the i-th *span* of the spline, and it represents the local interpolation of the function $\varphi(\cdot)$ on an interval spanned by four control points around the i-th one.

A very important evaluation for the algorithm derivation, is the derivative of (2) with respect to its local parameter u_n. It is easily evaluated in:

$$\varphi_i'(u_n) = \dot{\mathbf{u}}_n^T \mathbf{C} \mathbf{q}_{i,n}, \tag{3}$$

where $\dot{\mathbf{u}}_n \in \mathbb{R}^{4\times 1} = \left[3u_n^2,\, 2u_n,\, 1,\, 0\right]^T$.

The on-line learning algorithm can be derived by considering the cost function[1] (CF) $\hat{J}(\mathbf{w}_n, \mathbf{q}_{i,n}) = E\{e^2[n]\}$, where $e[n] = y[n] - d[n]$ is the error, $y[n]$ the system output and $d[n]$ the reference signal, usually corrupted with an additive noise $v[n]$. As usual, this CF is approximated by considering only the instantaneous error:

$$J(\mathbf{w}_n, \mathbf{q}_{i,n}) = e^2[n]. \tag{4}$$

The expression of the error signal $e[n]$ in (4) and the number of free parameters depend on the particular architecture used (Wiener, Hammerstein or Sandwich). Fig. 1 shows the WSAF, HSAF, S1SAF and S2SAF systems. The learning rules for the free parameters of the different SAF architectures are summarized in Table I. For all the details we refer to [8]–[10].

III. Combination of Wiener-Hammerstein SAF

Let us denote with $y_1[n], y_2[n], \ldots, y_N[n]$ the outputs of N Wiener, Hammerstein and/or Sandwich systems. We then form the output $y[n]$ of the combined architecture as:

$$y[n] = \sum_{k=1}^{N} h_k[n] y_k[n], \tag{5}$$

[1]In this work, we consider only real-valued variables.

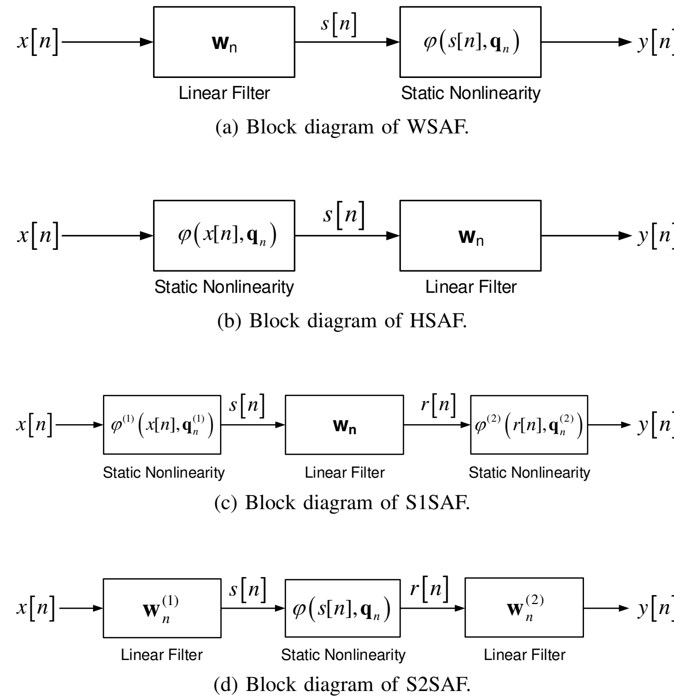

Fig. 1: Block diagrams of WSAF (a), HSAF (b), S1SAF (c) and S2SAF (d) systems.

where the $h_k[n]$ are N mixing parameters with values into the interval $[0, 1]$. Usually, a convex combination is preferred, hence $\sum_{k=1}^{N} h_k[n] = 1$. As a simple example, the convex combination of a Wiener and an Hammerstein SAF is shown in Fig. 2. The generalization to a combination of a greater number of SAFs is straightforward.

The output error of the whole architecture can then be written simply by:

$$e[n] = d[n] - y[n] = \sum_{k=1}^{N} h_k[n] e_k[n], \tag{6}$$

Architecture	**Learning rule**
WSAF	$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_w e[n] \varphi_i'(u) \mathbf{x}_n$ $\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_q e[n] \mathbf{C}^T \mathbf{u}$
HSAF	$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_w[n] e[n] \mathbf{s}_n$ $\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_q[n] e[n] \mathbf{C}^T \mathbf{U}_{i,n} \mathbf{w}_n$
S1SAF	$\mathbf{q}_{i,n+1}^{(2)} = \mathbf{q}_{i,n}^{(2)} + \mu_q^{(2)}[n] e[n] \mathbf{C}^T \mathbf{u}_n^{(2)}$ $\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_w[n] e[n] \varphi_i'(u_n^{(2)}) \mathbf{s}_n$ $\mathbf{q}_{i,n+1}^{(1)} = \mathbf{q}_{i,n}^{(1)} + \mu_q^{(1)}[n] e[n] \varphi_i'(u_n^{(2)}) \mathbf{C}^T \mathbf{U}_{i,n}^{(1)} \mathbf{w}_n$
S2SAF	$\mathbf{w}_{n+1}^{(2)} = \mathbf{w}_n^{(2)} + \mu_w^{(2)}[n] e[n] \mathbf{r}_n$ $\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_q[n] e[n] \mathbf{C}^T \mathbf{U}_{i,n} \mathbf{w}_n^{(2)}$ $\mathbf{w}_{n+1}^{(1)} = \mathbf{w}_n^{(1)} + \mu_w^{(1)}[n] e[n] \varphi_i'(u_n) \mathbf{X}_n \mathbf{w}_n^{(2)}$

TABLE I: Learning rules for the different SAF architectures. For details on parameters see [8]–[10].

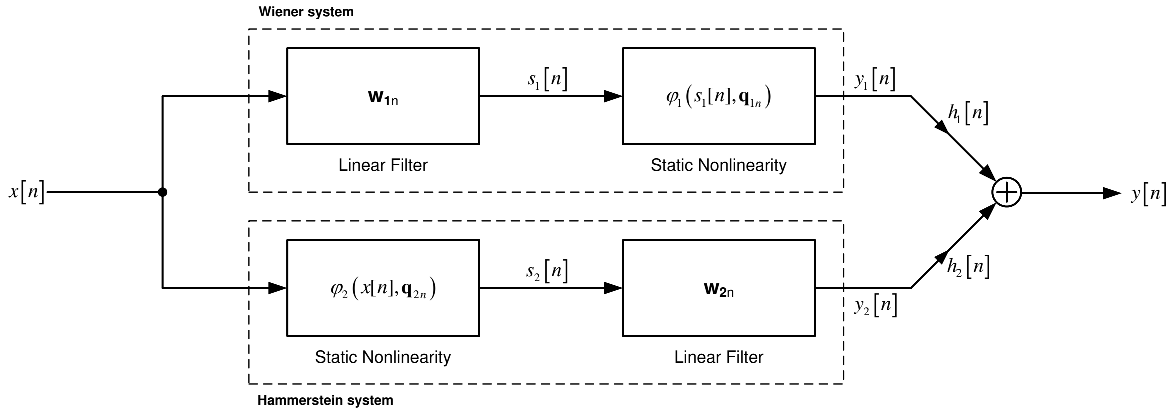

Fig. 2: Block diagram of the proposed combined Wiener-Hammerstein SAF architecture.

where the $e_k[n] = d[n] - y_k[n]$ are the single filter output errors and $d[n]$ is the reference signal, that is the output of the target model.

The adaptation of the mixing coefficients $h_k[n]$ are carried out by a constrained LMS (CLMS) algorithm (see for example [20]). In this regard, let us pose $\mathbf{h}_n = [h_1[n], h_2[n], \ldots, h_N[n]]^T$ the mixing coefficients vector at time n, then the adaptation is solved by the following minimization problem:

$$\min_{\mathbf{h}} \frac{1}{2} e^2[n], \quad \text{s.t.} \quad \mathbf{c}^T \mathbf{h}_n = b, \tag{7}$$

where $\mathbf{c} = \mathbf{1}_N \equiv [1, 1, \ldots, 1]^T$ and $b = 1$ in the considered case.

Problem (7) can be solved by the Lagrangian multipliers method. Specifically, the Lagrangian function $\mathcal{L}$ is the following:

$$\mathcal{L} = e^2[n] + \lambda \left(b - \mathbf{c}^T \mathbf{h}_n\right), \tag{8}$$

where λ is the Lagrangian multiplier. Taking the derivative of (8) with respect to the free parameters $\mathbf{h}_n$, after some mathematical manipulation, the solution of problem (7) is given by:

$$\mathbf{h}_{n+1} = \left\lfloor \mathbf{P}\left(\mathbf{h}_n + \mu_h[n] e[n] \mathbf{y}_n\right) + \mathbf{f} \right\rfloor_+, \tag{9}$$

where $\mathbf{P} = \mathbf{I} - \mathbf{c}\left(\mathbf{c}^T \mathbf{c}\right)^{-1} \mathbf{c}^T$, $\mathbf{f} = \mathbf{c}\left(\mathbf{c}^T \mathbf{c}\right)^{-1} b$ are a suitable matrix and vector that can be *a priori* computed, $\mu_h[n]$ is the learning rate and $\lfloor \delta \rfloor_+ = \max\{0, \delta\}$. This last operation is due to the sub-gradient projection method applied on the non-negative orthant [21], in order to fulfill the convexity constraints on each coefficient $h_k[n]$. In addition, the vector $\mathbf{y}_n = [y_1[n], y_2[n], \ldots, y_N[n]]^T$ collects the last output sample of the N filters. The algorithm is usually initialized by $\mathbf{h}_0 = \mathbf{f}$. Since the CLMS is well known in literature, we skip here the proof of (9), which can be found on many books like [20].

IV. Experimental results

In order to validate the proposed combination of nonlinear adaptive filters, some experimental results were performed. Experimental tests are addressed towards system identification problem in diverse and non-stationary scenarios.

A. First experiment

The experiment concerns the identification of an unknown combined system that consists in: (i) a Wiener filter composed by a linear FIR component $\mathbf{w}_0^W = [0.6, -0.4, 0.25, -0.15, 0.1, -0.05, 0.001]^T$ and a nonlinear memoryless target function implemented by a 23-points length LUT $\mathbf{q}_0^W$ and interpolated by a uniform third degree spline with an interval sampling $\Delta x = 0.2$ defined as:

$$\mathbf{q}_0^W = \{-2.2, -2.0, -1.8, \ldots, -1.0, -0.8, -0.91, -0.40, \\ -1.20, 0.05, 0.0, 0.90, 0.58, 1.0, 1.0, 1.2, 1.4, \ldots, 2.2\};$$

and, (ii) a Hammerstein system composed by the linear FIR component $\mathbf{w}_0^H = [0.3, -0.1, 0.7, -0.15, 0.1, -0.2, 0.01]^T$ and a nonlinear memoryless function implemented by a 23-points length LUT $\mathbf{q}_0^H$, defined as:

$$\mathbf{q}_0^H = \{-2.2, -2.0, -1.8, \ldots, -1.0, -0.8, -0.91, -0.20, \\ -0.40, 0.05, 0.0, 0.10, -0.30, 1.1, 1.0, 1.2, 1.4, \ldots, 2.2\}.$$

The input signal $x[n]$ consists in 50,000 samples of the signal generated by the following relationship:

$$x[n] = ax[n-1] + \sqrt{1-a^2}\,\xi[n], \tag{10}$$

where $\xi[n]$ is a zero mean white Gaussian noise with unitary variance and $0 \le a < 1$ is a parameter that determines the level of correlation between adjacent samples. Experiments were conducted with a set to 0.5. In addition it is considered an additive white noise $v[n]$ such that the signal to noise ratio is $SNR = 30$ dB. The whole system is non-stationary: we have split the experiment into four sections: every 12,500 samples the system to be identified is composed by different combination of the previous Wiener and Hammerstein systems. More specifically:

1) $1 \le n \le 12500$: only the Wiener system is active;
2) $12501 \le n \le 25000$: starting from an Hammerstein system, then it gradually fades towards a Wiener system;
3) $25001 \le n \le 37500$: both the system are simultaneous active (70% Hammerstein, 30% Wiener);
4) $37501 \le n \le 50000$: only the Hammerstein system is active.

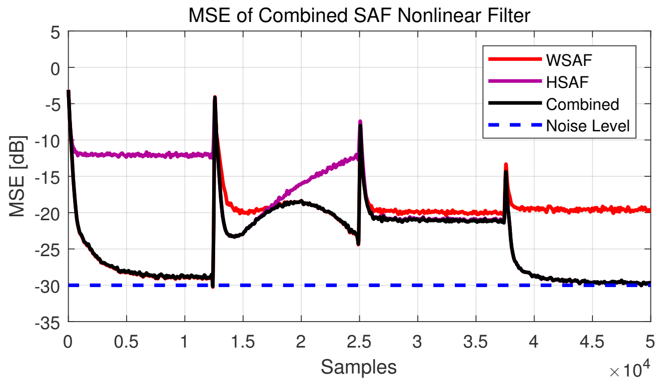

Fig. 3: MSE of the proposed convex combination of Wiener and Hammerstein SAFs.

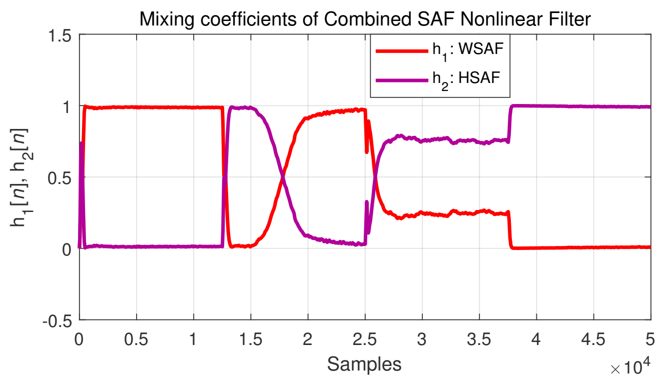

Fig. 4: Coefficients $h_1[n]$ and $h_2[n]$ of the proposed convex combination of Wiener and Hammerstein SAFs.

The learning rates are set to $\mu_w = \mu_q = 0.05$ for both the Wiener and Hammerstein architectures, while $\mu_h = 1$. Both the linear components use a filter length of $M = 15$. The filter weights are initialized as a Dirac impulse function, while both the nonlinearities have been initialized as a straight line. The mixing coefficients vector is initialized to $\mathbf{h}_0 = [1, 1]/2$.

Results in terms of MSE are reported in Fig. 3, while Fig. 4 shows the mixing coefficients $h_1[n]$ and $h_2[n]$, respectively. The reported results have been averaged over 100 independent runs. Both the figures confirm the effectiveness of the proposed approach: in the first tract the combination selects the WSAF; in the second tract, after the identification of the Hammerstein model, the combination gradually selects the Wiener one according to the model to be identified; in the third tract, both system are selected. An examination of Fig. 4 reveals that the combination consists in a 30% of a Wiener model and a 70% of an Hammerstein one, according to the model to be identified; finally, in the fourth tract, the combination selects the Hammerstein model. By looking at the transient tracts in Fig. 3, we can conclude that the convergence speed is fast and the tracking behavior of the proposed system is very good. However, from Fig. 3 it should be noted that the MSE is slightly worst when both the systems are simultaneously active.

B. Second experiment

In a second experimental set-up, there is a non-stationary target system that switches among three separate components: a Wiener model, a LNL sandwich model and a NLN sandwich one, that works separately in time.

The Wiener system consists in the cascade of an FIR linear filter represented by the following transfer function:

$$H_W(z) = 0.5-0.1z^{-1}+0.4z^{-2}-0.3z^{-3}+0.1z^{-4}-0.002z^{-5}, \tag{11}$$

and the following nonlinear function:

$$d[n] = G\left[\frac{1}{1+e^{-\alpha s[n]}} - 0.5\right], \tag{12}$$

where $G = 2$ and α assumes the value 4 if $s[n] > 0$, and the value 0.5 if $s[n] \leq 0$.

The LNL model (sandwich model 2) consists in the cascade of the linear filter $H_W(z)$ in (11), the nonlinear function in (12), and the following FIR linear filter:

$$H_{S2}(z) = 0.3-0.1z^{-1}+0.7z^{-2}-0.15z^{-3}+0.1z^{-4}-0.2z^{-5}, \tag{13}$$

Finally, the NLN model (sandwich model 1), drawn from [22], consists in the cascade of the following non-polynomial nonlinearity:

$$s[n] = \frac{x[n]}{\sqrt{0.9x^2[n]+0.1}}, \tag{14}$$

the following IIR linear filter:

$$H_{S1}(z) = 0.3\,\frac{0.5z^{-1}+0.25z^{-2}}{1-1.5z^{-1}+0.7z^{-2}}, \tag{15}$$

and the following polynomial nonlinear function:

$$d[n] = r[n] + 0.2\,r^3[n]. \tag{16}$$

The system used for the identification consists in the convex combination of the following three SAFs: a WSAF, and two sandwiches models (an S1SAF and an S2SAF, respectively). The mixing coefficients associated to these systems are $h_1[n]$, $h_2[n]$ and $h_3[n]$, respectively. Again, the filter weights are initialized as a Dirac impulse function, while both the non-linearities have been initialized as a straight line. In addition, the mixing coefficients are initialized to $\mathbf{h}_0 = [1, 1, 1]/3$.

Experiments were conducted with the input signal $x[n]$ in (10) with $a = 0.1$ and an $SNR = 30$ dB. A total of 60,000 samples have been used, and the three systems are switched every 20,000 samples.

Results in terms of MSE are reported in Fig. 5, while Fig. 6 shows the mixing coefficients $h_1[n]$, $h_2[n]$ and $h_3[n]$, respectively. The reported results have been averaged over 100 independent runs. These figures show that the convex optimization is able to correctly identify the related target model: in the first tract the combination selects the WSAF model; in the second tract the combination selects the S2SAF models; finally, in the third tract the S1SAF has been selected. This behavior is clearly visible in Fig. 6, but it is also evident in Fig. 5 from the numerical values of the related MSEs. This

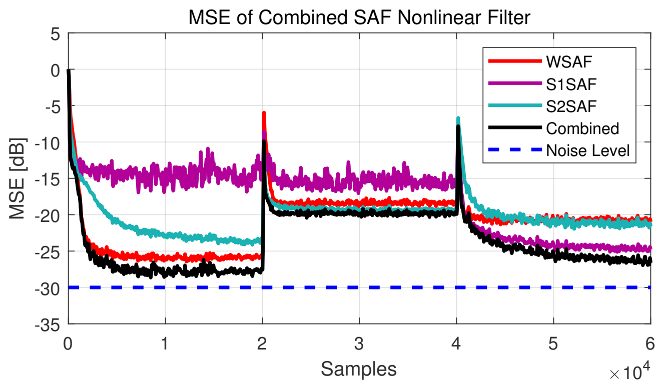

Fig. 5: MSE of the proposed convex combination of three SAFs.

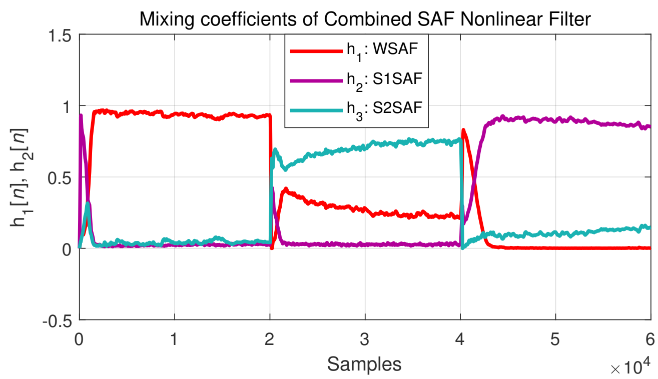

Fig. 6: Coefficients $h_1[n]$, $h_2[n]$ and $h_3[n]$ of the proposed convex combination of three SAFs.

last figure shows also that, although all components of the system have been correctly identified, the MSE of the S2SAF provides a value greater than the other models. In addition, Fig. 5 also shows that the S1SAF model cannot work appropriately in the first two tracts, this because here the first subsystem of the target is always a linear component. By looking at the transient tracts in Fig. 5, we can conclude that the convergence speed is fast and the tracking behavior of the proposed system is very good also in this case.

As a final remark, let us note that the MSE of the convex combination of the three used models always provides a total error that is less than the single errors.

V. Conclusion

In this paper we have presented a convex combination of nonlinear spline adaptive filters (SAFs). The system implemented is based on Wiener, Hammerstein and Sandwiches architectures, and it is adapted by using a constrained LMS (CLMS) approach. The learning rule for the mixing coefficients has been analytically derived. The convex combination of SAFs is able to work in a non-stationary environment and without an explicit *a priori* knowledge of the model to be identified. Some experimental tests have demonstrated the effectiveness of the proposed approach. In particular, the obtained results show that the nonlinear models have been always correctly identified, the convergence speed is fast and the tracking behavior of the proposed approach is very good.

References

[1] D. Comminiello and J. C. Principe, Eds., *Adaptive Learning Methods for Nonlinear System Modeling*. Butterworth-Heinemann, 2018.

[2] S. Haykin, *Neural Networks and Learning Machines*, 2nd ed. Pearson Publishing, 2009.

[3] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*. Wiley, 2010.

[4] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*. Wiley, 2000.

[5] L. Ljung, *System Identification - Theory for the User*, 2nd ed. Upper Saddle River, 1999.

[6] R. Haber and H. Unbehauen, "Structure identification of nonlinear dynamic systems — A survey on input/output approaches," *Automatica*, vol. 26, no. 4, pp. 651–677, 1990.

[7] L. Ljung, "Identification of nonlinear systems," in *9th International Conference on Control, Automation, Robotics and Vision (ICARCV 2006)*, Singapore, 2006.

[8] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Nonlinear spline adaptive filtering," *Signal Processing*, vol. 93, no. 4, pp. 772–783, April 2013.

[9] ——, "Hammerstein uniform cubic spline adaptive filters: Learning and convergence properties," *Signal Processing*, vol. 100, pp. 112–123, July 2014.

[10] ——, "Novel cascade spline architectures for the identification of nonlinear systems," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. 62, no. 7, pp. 1825–1835, July 2015.

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

[12] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, March 2006.

[13] J. Arenas-García, L. A. Azpicueta-Ruiz, M. T. M. Silva, V. H. Nascimento, and A. H. Sayed, "Combinations of adaptive filters: Performance and convergence properties," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 120–140, January 2016.

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

[15] M. T. Akhtar, "A convex-combined step-size-based normalized modified filtered-x least mean square algorithm for impulsive active noise control systems," in *26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, September 2018, pp. 2454–2458.

[16] B. K. Das, S. Mukhopadhyay, and M. Chakraborty, "Robust adaptive filtering via convex combination of ℓ_0-RLS adaptive filters," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS 2018)*, Florence, Italy, May 2018, pp. 1–5.

[17] D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-García, and A. Uncini, "Functional link adaptive filters for nonlinear acoustic echo cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1502–1512, July 2013.

[18] Y. Zheng, J. Dong, W. Ma, and B. B. Chen, "Kernel adaptive Hammerstein filter," in *26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, September 2018, pp. 504–508.

[19] Y. Chien and J. Li-You, "Convex combined adaptive filtering algorithm for acoustic echo cancellation in hostile environments," *IEEE Access*, vol. 6, pp. 16 138–16 148, 2018.

[20] A. Uncini, *Fundamentals of Adaptive Signal Processing*. Springer, 2015.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[22] Y. Zhu, "Estimation of an N-L-N Hammerstein-Wiener model," *Automatica*, vol. 38, no. 9, pp. 1607–1614, September 2002.