A Priority-Based Collaborative FLAF Exploiting Different Functional Expansions

Danilo Comminiello

Abstract

This document introduces a collaborative architecture for nonlinear adaptive filtering. This system is based on the joined use of different *functional link adaptive filters* (FLAFs), each one defined by a different choice of the functional expansion type. The FLAFs involved work in parallel according to a collaborative approach, thus the architecture considers the output of each filter only when it actively contributes to the overall nonlinear modelling. Moreover, the FLAFs are sorted according to an order of priorities, which facilitates and biases the collaboration between the nonlinear models.

I. Introduction

Identification of nonlinear systems has always drawn great attention of scientific research, also due to the wide variety of real-world nonlinear processes [1]. A multiplicity of solutions have been proposed to address this problem, which has revealed itself as meeting point between machine learning and signal processing.

A promising nonlinear model is the *functional link adaptive filter* (FLAF) [2], [3]. 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 [4]. Efficient FLAF-based architectures are introduced in [2], thus representing an effective solution to model nonlinearities in applications like NAEC, in which the separation of linear and nonlinear filterings leads to a performance improvement. One of the main advantages of such architectures lies in the flexibility, since the setting of several parameters is allowed in order to fit the model to a specific application. In this regard, an important choice in the FLAF design concerns the expansion type, i.e. the basis functions, or a subset of it, to assign for each functional link. This choice mostly depends on the application and on the signals involved in the processing. Basis functions must satisfy universal approximation constraints and may be a subset of orthogonal polynomials, such as Chebyshev, Legendre and trigonometric polynomials, or just approximating functions, such as sigmoid and Gaussian functions. Functional expansions based on Chebyshev polynomials have been widely used for several applications, in particular for nonlinear dynamic system identification [5] and channel equalization [6], showing a strong effectiveness. Legendre functional links was used as an alternative to Chebyshev polynomials in channel equalization [7], but the most popular functional link sets are derived from trigonometric basis expansion. In particular, trigonometric polynomials were used for different applications, from function approximation [4], [8], to active noise control [3] and NAEC [2].

On the basis of the foregoing statements, when some *a priori* information is available regarding the kind of nonlinearities to model, it is more easy to choose the functional expansion and design an efficient FLAF. However, in many cases, the kind of nonlinearity may vary in time due to any alteration of the environment, and therefore it is very difficult coming to an appropriate choice. In order to address this problem, we introduce a collaborative FLAF-based architecture involving different FLAFs, each one characterized by a different functional expansion. At each time instant, the architecture evaluates individually the outputs of FLAFs, according to any minimization criterion of the mean square error, and takes them into account only if they actively contributes to the overall nonlinear modelling. Such FLAF-based architecture draws inspiration from recently proposed works on the adaptive combination of filters [9], [10] and, in particular, on the collaboration of adaptive filters [2], [11], [12]. Differently from other collaborative models, the proposed architecture allows an exchange of information between the involved FLAFs based on an order of priorities. The resulting model is efficiently capable of modelling nonlinearities regardless of their nature and the unknown system to identify.

II. Nonlinear Modelling Using Functional Link Adaptive Filters

In this section we present a brief overview of the FLAF model, that we use to devise the proposed architecture.

A. The Functional Link Adaptive Filter Model

The *functional link adaptive filter* (FLAF) model is based on the representation of the input signal in a higher-dimensional space [4], in which an enhanced nonlinear modelling is allowed. Such approach derives from the machine learning theory, more precisely from the Cover's Theorem on the separability of patterns.

The FLAF is composed of two main parts: a nonlinear *functional expansion block* (FEB) and a subsequent linear 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 $\Phi = \left\{\varphi_0(\cdot), \varphi_1(\cdot), \ldots, \varphi_{Q-1}(\cdot)\right\}$, where Q is the number of functional links. At the n-th time instant, the FEB receives the input sample $x[n]$, which is stored in an input buffer $\mathbf{x}_{\mathrm{F},n} \in \mathbb{R}^{M_\mathrm{i}} = \left[\begin{array}{cccc} x[n] & x[n-1] & \ldots & x[n-M_\mathrm{i}+1] \end{array}\right]^T$, where M_i is defined as the input buffer length. Each element of $\mathbf{x}_{\mathrm{F},n}$ is passed as argument to the chosen set of functions Φ, thus yielding a subvector $\overline{\mathbf{g}}_{i,n} \in \mathbb{R}^Q$:

$$\overline{\mathbf{g}}_{i,n} = \left[\begin{array}{c} \varphi_0(x[n-i]) \\ \varphi_1(x[n-i]) \\ \vdots \\ \varphi_{Q-1}(x[n-i]) \end{array}\right]. \tag{1}$$

The concatenation of all the subvectors, for $i = 0, \ldots, M_\mathrm{i}-1$, engenders an *expanded buffer* $\mathbf{g}_n \in \mathbb{R}^{M_\mathrm{e}}$:

$$\begin{aligned} \mathbf{g}_n &= \left[\begin{array}{cccc} \overline{\mathbf{g}}_{0,n}^T & \overline{\mathbf{g}}_{1,n}^T & \cdots & \overline{\mathbf{g}}_{M_\mathrm{i}-1,n}^T \end{array}\right]^T \\ &= \left[\begin{array}{cccc} g_0[n] & g_1[n] & \ldots & g_{M_\mathrm{e}-1}[n] \end{array}\right]^T \end{aligned} \tag{2}$$

where $M_\mathrm{e} \geq M_\mathrm{i}$ represents the length of the expanded buffer. Note that $M_\mathrm{e} = M_\mathrm{i}$ only when $Q = 1$.

The achieved expanded buffer $\mathbf{g}_n$ is then fed into a linear adaptive filter $\mathbf{w}_{\mathrm{F},n} \in \mathbb{R}^{M_\mathrm{e}} = \left[\begin{array}{ccc} w_{\mathrm{F},0}[n] & w_{\mathrm{F},1}[n] & \ldots \end{array}\right.$ $\left. w_{\mathrm{F},M_\mathrm{e}-1}[n]\right]^T$, thus providing the nonlinear output:

$$y_\mathrm{F}[n] = \mathbf{g}_n^T \mathbf{w}_{\mathrm{F},n-1}. \tag{3}$$

Thereby, the nonlinear error signal is:

$$e_\mathrm{F}[n] = d_\mathrm{F}[n] - y_\mathrm{F}[n] \tag{4}$$

which is used for the adaptation of $\mathbf{w}_{\mathrm{F},n}$. In (4), $d_{\mathrm{F},n}[n]$ represents the desired signal for the nonlinear model. Being $\mathbf{w}_{\mathrm{F},n}$ a conventional linear filter, it can be adapted by any adaptive algorithm based on the minimization of the mean square error.

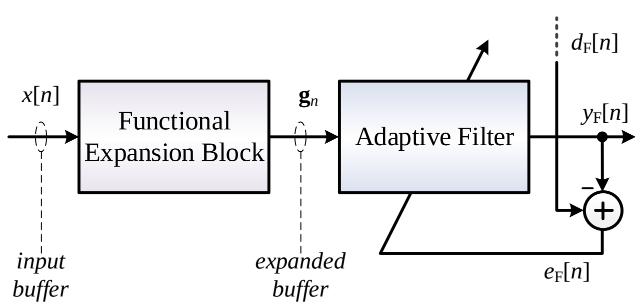

Fig. 1. The functional link adaptive filter.

B. Functional Expansion Types

In the FLAF model described above, a fundamental role is played by the FEB, therefore, the choice of the set Φ of functional links represents a crucial decision. This choice may depend on the kind of application and on the kind of signals involved in the processing. The functional links must satisfy the universal approximation property, and they may be a subset of orthogonal polynomials or simple approximating functions. Here we introduce the most used functional link sets. Note that in all the sets we consider only nonlinear elements, such that every processing on the linear elements is demanded to a separate linear adaptive filter [2].

1) Chebyshev polynomial expansion: Chebyshev polynomial functions are endowed with powerful nonlinear approximation capability. This is the reason why their use is widespread in different fields of application. In [5], [6], it was pointed out that an artificial nural network (ANN) with Chebyshev polynomial expansion has universal approximation capability and faster convergence than a multi-layer perceptron (MLP) network. The effectiveness of Chebyshev polynomial expansion is mainly due to the fact that it includes functions of previously computed functions. This is the reason why its use results successful especially for the identification of nonlinear dynamic systems [5]. Moreover, Chebyshev expansion is based on power series expansion, which may approximate a nonlinear function with a very small error near the point of expansion. However, far from the point of expansion, the error may increases rapidly. With reference to other power series of the same degree, Chebyshev polynomials are quite computationally cheap and more efficient. However, when the power series converges slowly the computational cost dramatically increases.

Taking into account the i-th input sample $x[n-i]$, the Chebyshev polynomial expansion can be written as:

$$\begin{aligned}\varphi_j\left(x[n-i]\right) = 2x[n-i]\,\varphi_{j-1}\left(x[n-i]\right)\\ -\varphi_{j-2}\left(x[n-i]\right)\end{aligned} \tag{5}$$

for $j = 0, \ldots, P-1$, where P is the *expansion order*. Note that the number of functional links in a Chebyshev set is equal to the expansion order, i.e. $Q = P$. In (5), initial values (for $j = 0$) are:

$$\begin{aligned}&\varphi_{-1}\left(x[n-i]\right) = x[n-i]\\ &\varphi_{-2}\left(x[n-i]\right) = 1.\end{aligned} \tag{6}$$

2) Legendre polynomial expansion: Similar to Chebyshev polynomials, the Legendre functional links provide computational advantage while promising better performance [7]. Legendre polynomial expansions have been used for applications like channel equalization [7], in which Legendre-based *quadrature amplitude modulation* (QAM) equalizer performs better than radial basis function (RBF)-based and linear FIR-based equalizers.

Considering the i-th input sample, the Legendre polynomial expansion is described by:

$$\begin{aligned}\varphi_j\left(x[n-i]\right) = \frac{1}{j}\Big\{(2j-1)\,x[n-i]\,\varphi_{j-1}\left(x[n-i]\right)\\ -(j-1)\,\varphi_{j-2}\left(x[n-i]\right)\Big\}\end{aligned} \tag{7}$$

for $j = 0, \ldots, P-1$. As for the Chebyshev expansion, the number of functional links is $Q = P$. Also in (7), initial values are set as (6).

3) Trigonometric series expansion: When trigonometric polynomials are used in upstream, i.e. before the adaptive filtering, the weight estimate will approximate the desired impulse response in terms of multidimensional Fourier series decomposition. In particular, compared with other orthogonal basis functions, trigonometric polynomials provide the best compact representation of any nonlinear function in the mean square sense, even for nonlinear dynamic systems as proved in [8]. Moreover, trigonometric functions are computationally cheaper than power series-based polynomials. Due to its properties, trigonometric series expansion is very popular in several applications [2]–[4], [8].

Taking into account the i-th sample of the input buffer, it is possible to generalized the set of functional links using trigonometric basis expansion as:

$$\varphi_j\left(x[n-i]\right) = \begin{cases}\sin\left(p\pi x[n-i]\right), & j = 2p-2\\ \cos\left(p\pi x[n-i]\right), & j = 2p-1\end{cases} \tag{8}$$

where $j = 0, \ldots, Q-1$ is the functional link index and $p = 1, \ldots, P$ is the expansion index, being P the *expansion order*. In this case, the functional link set is composed of $Q = 2P$ functional links. Note that the expansion order for the trigonometric series is different from the order of both Chebyshev and Legendre polynomials. Equation (8) describes a memoryless expansion. A trigonometric expansion with memory can be easily achieved by adding the cross-product terms, as detailed in [2]. It may be very useful in the modelling of dynamic nonlinearities. In a trigonometric expansion with memory, the *memory order* K determines the length of the additional functional links, i.e., the depth of the outer products between the i-th input sample and the functional links related to the previous input samples.

III. A Priority-Based CFLAF Architecture with multiple expansions

The choice of the functional expansion type is easy when *a priori* information on signals to process and on the scenario are available. However, very often there is no *a priori* knowledge in that regard and also, the scenario may vary in time. In such cases, a particular choice of the functional link set may not result appropriate. In order to address this problem we propose a collaborative architecture which involves different functional expansion types and exploits their properties, thus being appropriate for the modelling of several nonlinear scenarios.

The collaborative approach of the proposed architecture derives from [2], in which the *collaborative functional link adaptive filter* (CFLAF) was proposed to address a serious problem in NAEC. More precisely, when the nonlinearity level affecting the microphone signal is negligible, the nonlinear filter only brings some gradient noise in the filtering process, thus degrading NAEC performance more than a linear filter. On the other hand, the CFLAF is capable of activating or deactivating automatically the nonlinear filtering path, thus avoiding any performance decrease.

The proposed architecture, depicted in Fig. 2, is an expansion of the CFLAF in [2], involving different nonlinear FLAF, each one defined by a different functional expansion type. As it is possible to see from Fig. 2, the overall output signal $y[n]$ of the architecture is composed of a linear contribution $y_{\mathrm{L}}[n]$ and a nonlinear one $y_{\mathrm{N}}[n]$. In particular, $y_{\mathrm{N}}[n]$ results from the weighted sum of the individual FLAF outputs, i.e.:

$$\begin{aligned} y[n] &= y_{\mathrm{L}}[n] + y_{\mathrm{N}}[n] \\ &= y_{\mathrm{L}}[n] + \mathbf{h}_n^T \mathbf{y}_{\mathrm{F}i,n} \end{aligned} \tag{9}$$

where $\mathbf{h}_n \in \mathbb{R}^L = \begin{bmatrix} h_1[n] & h_2[n] & \dots & h_L[n] \end{bmatrix}^T$ is the vector of the *shrinkage parameters*, being L the number of nonlinear path involved by the architecture. Each shrinkage parameter aims at weighting the output of the corresponding nonlinear FLAF, thus allowing it to be added or not to the overall output signal. Each nonlinear path of the architecture, including the corresponding shrinkage parameter, can be considered as a convex combination between the nonlinear FLAF and a null *virtual* filter, i.e. an *all-zero kernel*, whose coefficients are static and set to zero [11].

A distinctive feature of the proposed architecture lies in the fact that the nonlinear filtering paths are chosen and sorted according to an order of priorities. In this way, the architecture "will ask" a possible nonlinear contribution to the first FLAF and then to the other ones according to the given order. Such order is driven by the desired signal received by each FLAF, which is progressively and possibly refined by the previous FLAFs. This also implies a "collaboration" between each filter. Since the collaboration follows a given order of priorities, we refer to the proposed scheme as *priority-based CFLAF* (PBCFLAF).

A consequence of the priority-based collaboration is that each filter is adapted by using a different error signal. In particular, the linear filter $\mathbf{w}_{\mathrm{L},n}$ pursues the minimization of the overall error signal, i.e. $e[n] = d[n] - y[n]$, since the output contribution of the linear filter is always present when there is an impulse response to estimate. However, the coefficients of the filters on the nonlinear paths $\mathbf{w}_{\mathrm{F}i,n}$ are updated by using their own local error signals $e_{\mathrm{F}i}[n]$, which are free from the output contribution of the previous filters:

$$e_{\mathrm{F}i}[n] = d[n] - y_{\mathrm{L}}[n] - \sum_{k=1}^{i} y_{\mathrm{F}k}[n] \tag{10}$$

for $i = 1, \dots, L$.

The shrinkage parameters $h_i[n]$ are adapted separately from each other, subject to convex constraints $0 \le h_i[n] \le 1$, through the adaptation of auxiliary parameters $a_i[n]$, which related to $h_i[n]$ by means of a sigmoid function [10]:

$$h_i[n] = \mathrm{sgm}(a_i[n]) = \frac{1}{\left(1 + e^{-a_i[n]}\right)}. \tag{11}$$

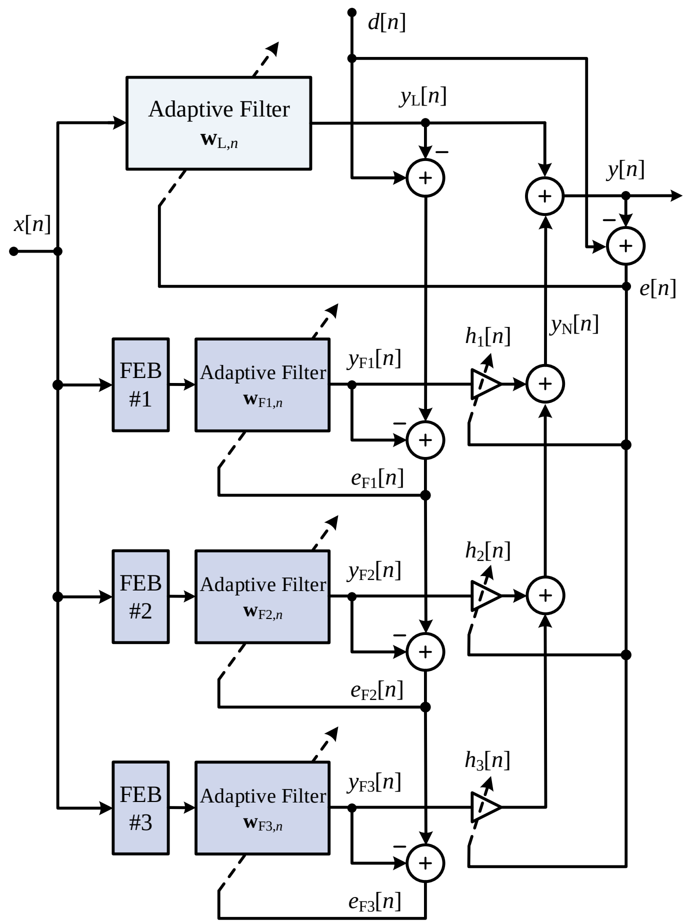

Fig. 2. Priority-based CFLAF architecture involving three different nonlinear FLAFs.

The weights $h_i\left[n\right]$ are computed by adapting the auxiliary parameters according to a gradient descent rule:

$$\begin{aligned} a_i\left[n+1\right] &= a_i\left[n\right] - \frac{1}{2}\mu_a \frac{\partial e^2\left[n\right]}{\partial a_i\left[n\right]} \\ &= a_i\left[n\right] + \frac{\mu_a}{r_i\left[n\right]} e\left[n\right] y_{\mathrm{F}i}\left[n\right] h_i\left[n\right] \left(1 - h_i\left[n\right]\right) \end{aligned} \tag{12}$$

where μ_a is a step size parameter, whose value is the same for all the FLAFs, and

$$r_i\left[n\right] = \beta r_i\left[n-1\right] + \left(1-\beta\right) y_{\mathrm{F}i}^2\left[n\right] \tag{13}$$

is a rough low-pass filtered estimate of the power of the signal of interest. The parameter β is a smoothing factor which ensures that $r_i\left[n\right]$ is adapted faster than any filter component. The value of $a_i\left[n\right]$ is kept within $\left[4, -4\right]$ for practical reasons [10].

The described PBCFLAF is robust against different kinds of nonlinearity, even in changing environments. According to the chosen priority, when an FLAF is necessary for the nonlinear modelling, the related shrinkage parameter is close to 1. On the other hand, when the output of an FLAF does not contribute to model a nonlinear signal, the related weight approaches 0. Note that when the scenario is merely linear, the PBCFLAF takes into account only the output of the linear filter, i.e. $h_i\left[n\right] = 0$ for $i = 1, \ldots, L$.

REFERENCES

[1] L. Ljung, *System Identification – Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[2] D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-García, and A. Uncini, "Functional link adaptive filters for nonlinear acoustic echo cancellation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 7, pp. 1502–1512, Jul. 2013.

[3] G. L. Sicuranza and A. Carini, "A generalized FLANN filter for nonlinear active noise control," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2412–2417, Nov. 2011.

[4] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.

[5] J. C. Patra and A. C. Kot, "Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 4, pp. 505–511, Aug. 2002.

[6] H. Zhao and J. Zhang, "Functional link neural network cascaded with Chebyshev orthogonal polynomial for nonlinear channel equalization," *Signal Process.*, vol. 88, no. 8, pp. 1946–1957, Aug. 2008.

[7] J. C. Patra, W. C. Chin, P. K. Meher, and G. Chakraborty, "Legendre-FLANN-based nonlinear channel equalization in wireless communication system," in *Proc. of IEEE Int. Conf. Syst., Man Cybern. (SMC '08)*, Singapore, Oct. 2008, pp. 1826–1831.

[8] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, pp. 254–262, Apr. 1999.

[9] J. Arenas-García, M. Martínez-Ramón, A. Navia-Vázquez, and A. R. Figueiras-Vidal, "Plant identification via adaptive combination of transversal filters," *Signal Process.*, vol. 86, no. 9, pp. 2430–2438, Sep. 2006.

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

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

[12] D. P. Mandic, P. Vayanos, C. Boukis, B. Jelfs, S. Goh, T. Gautama, and T. Rutkowski, "Collaborative adaptive learning using hybrid filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '07)*, vol. 3, Honolulu, HI, 2007, pp. 921–924.