8

Functional Link Adaptive Filters: A New Class of Nonlinear Filters

Contents

THIS chapter introduces a new class of nonlinear filters, whose structure is based on Hammerstein model. The *functional link adaptive filters* (FLAF) are defined by a nonlinear input expansion, which enhances the representation of the input signal through a projection in a higher dimensional space, and a subsequent linear filtering. The most important element of a functional link adaptive filter is the nonlinear expansion, in which the a set of *functional links* processes the input signal allowing an enhanced modelling of nonlinearities. The functional expansion block allows to design a suitable filter according to scopes and field of application. This flexibility enables the filter to find the optimal trade-off between performance and computational complexity, according to the specifications of the problem.

8.1 INTRODUCTION

The problem of modelling linear systems has been widely tackled in last decades [116, 69] and, nowadays, it may be considered definitely solved. A linear system can be considered as a *white box*, since all information necessary to describe the system is available. Therefore, an effective estimate of the impulse response of a linear system may be achieved by using linear adaptive filtering algorithms [120, 59]. However, real-world systems often involve some degree of nonlinearity. In particular, if a system introduces a weak degree of nonlinearity it can be considered as a *grey box*, since, although information concerning the system is not entirely known, a linear approximation may be adopted. However, if a system shows a strong degree of nonlinearity it can be considered as a *black box*, since no information concerning the system is *a priori* available, thus a nonlinear system identification technique must be taken into account [157].

A popular approach to the problem of nonlinear system identification is the use of a cascade of a linear dynamic system and a memoryless nonlinear function. This kind of model is known in literature as *Wiener model* [97, 157]. On the other side, a cascade of a memoryless nonlinear function and a linear dynamic system is a very useful system in many practical applications and it is known as *Hammerstein model* [97]. Among the several other solutions to nonlinear filtering problem, one of the most popular technique proposed in literature is based on the so-called *polynomial filters* [86], which is a quite general model for nonlinear filtering. In this kind of filters, the adaptive nonlinearity consists in a polynomial-type nonlinearity: the filter output can be evaluated from its input through a polynomial model, truncated to a suitable order.

A particular case of polynomial filters is represented by Volterra filters [150]. The Volterra model can be very effectiveness in many practical applications, however, as said in Section 7.1, its computational cost may be very huge due the enormous number of coefficients required for higher-order kernels.

A more general framework for nonlinear filtering is provided by *artificial neural networks* (ANNs) [60], which represent an easily and flexible way to implement a such nonlinear filtering. The nonlinear transformations, applied by each neuron of an ANN, realize the searched nonlinearity. ANNs are capable of generating complex mapping between input and output space, therefore, arbitrarily complex nonlinear decision boundaries can be approximated by these networks. A drawback of this approach is the high computational cost of such a network. A particular type of ANN with reduced computational cost is characterized by activation functions implemented as *flexible spline nonlinear functions*, which are piecewise polynomials [148, 55, 129]. The term *spline*, in fact, comes from the flexible spline devices used by drafters to draw smooth shapes. Such networks, due to the adaptability of their activation functions, can solve hard problems with a low number of neurons [114].

In this chapter, we propose a novel nonlinear adaptive filtering model based on *functional links*. 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 initially proposed by Pao [103] with the aim of developing a class of single-layer feedforward neural networks, known as *functional link artificial neural networks* (FLANNs). Pao has shown that FLANN may be conveniently used for function approximation and pattern recognition with faster convergence rate and lesser computational load than a *multi-layer perceptron* (MLP) ANN [103]. The FLANN is basically a flat net and the removal of the hidden layer allows a very simple use of the *backpropagation* learning algorithm [103, 60]. Functional links have been used for many applications, ranging from pattern recognition [104] to process control [128].

In this research study we develop a novel nonlinear model based on functional links that is not built on an ANN but on an adaptive filter structure. Such model, named *functional link adaptive filter* (FLAF), exploits the nonlinear modelling capabilities of functional links and the filtering properties of linear adaptive algorithms, which are definitely less computationally expensive than ANNs, thus resulting an effective tool to model nonlinearities (especially) in acoustic applications.

8.2 NONLINEAR SYSTEM IDENTIFICATION PROBLEM

Before describing the proposed nonlinear model, we briefly introduce a problem formulation concerning the nonlinear system identification. It needs to notice that the correspondent acoustic application of nonlinear system identification is the nonlinear acoustic echo cancellation, that we address in the next chapter.

A nonlinear system identification problem based on a Hammerstein model is depicted in Fig. 8.1, in which it is possible to notice that the desired sig-

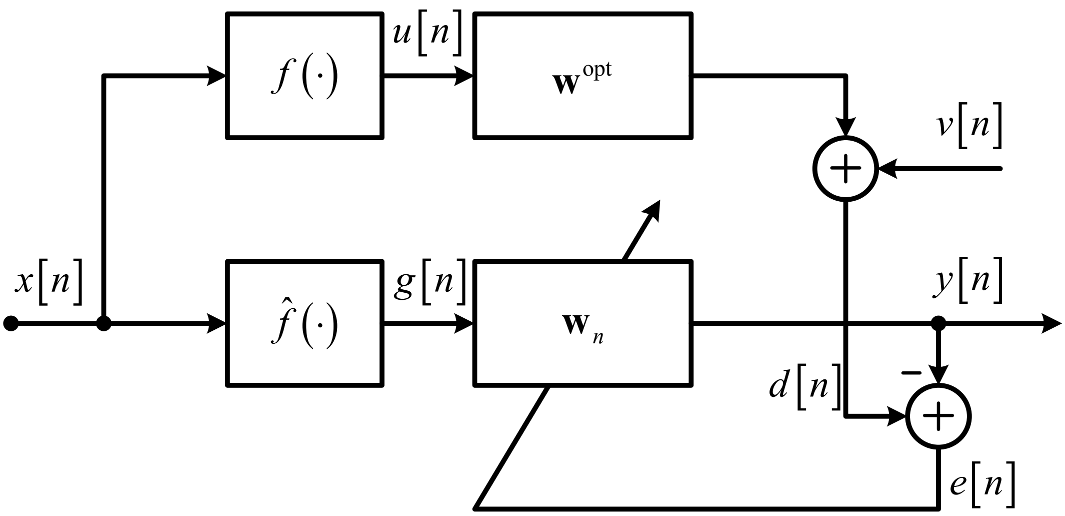

Fig. 8.1: *Hammerstein-based nonlinear system identification scheme.*

nal $d[n]$ results from the convolution between the input signal $x[n]$ and the unknown system to identify, denoted as:

$$\mathbf{w}^{\text{opt}} = \left(\mathbf{x}_n^T \mathbf{x}_n\right)^{-1} \mathbf{x}_n d[n] \tag{8.1}$$

as it is the optimal solution that solves the least-mean squares problem:

$$\min_{\mathbf{w}} \mathrm{E}\left\{\left|d[n] - \mathbf{x}_n^T \mathbf{x}_{n-1}\right|^2\right\}. \tag{8.2}$$

In a Hammerstein model the system to identify is preceded by a nonlinearity which is *a priori* unknown and may only be approximated. Therefore the identification of a Hammerstein model strictly depends on the nonlinearity upstream the filter.

In Fig. 8.1 it is possible to notice that the signal $x[n]$ is fed into a nonlinear system, thus the input signal to the unknown system gets to be $u[n] = f(u[n])$. Therefore the desired signal is:

$$d[n] = \mathbf{u}_n^T \mathbf{w}^{\text{opt}} + v[n] \tag{8.3}$$

where $v[n]$ is an additive noise, usually a white Gaussian noise with zero mean and unitary variance, thus resulting independent and identically distributed (i.i.d.). Consequently, the adaptive nonlinear filter, that aims at identifying the unknown system, is composed of a linear adaptive algorithm preceded by an artificial nonlinearity $\hat{f}(\cdot)$, which aims at approximating the nonlinearity of the unknown system. Therefore, the nonlinear input to the linear adaptive filter is denoted as $g[n] = \hat{f}(x[n])$.

The scheme depicted in Fig. 8.1 is generic for a system identification problem based on a Hammerstein model; with some specific changes, it allows to analyze a wide class of adaptive nonlinear filters based on Hammerstein model and described by the following adaptation rule:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \mathbf{g}_n \gamma(e[n]) \tag{8.4}$$

where $\gamma(\cdot)$ represents some function of the *a priori* output error signal:

$$e[n] = d[n] - \mathbf{g}_n^T \mathbf{w}_{n-1}. \tag{8.5}$$

Therefore, the scope is to define a suitable nonlinear function $\hat{f}(\cdot)$, which allows, through the update of an adaptive filter $\mathbf{w}_n$, to minimize the mean square error.

8.3 FUNCTIONAL LINK ADAPTIVE FILTERS

8.3.1 Functional link approach

The main idea which underpins our FLAF approach is that of asking whether it might be possible to enhance the original representation right from the start in a linearly independent manner. A way of enhancing the original input signal is to represent it in a space of higher dimension [103]. This process derives directly from the machine learning theory, and more exactly from Cover's Theorem on the separability of patterns [60]. Size and nature of

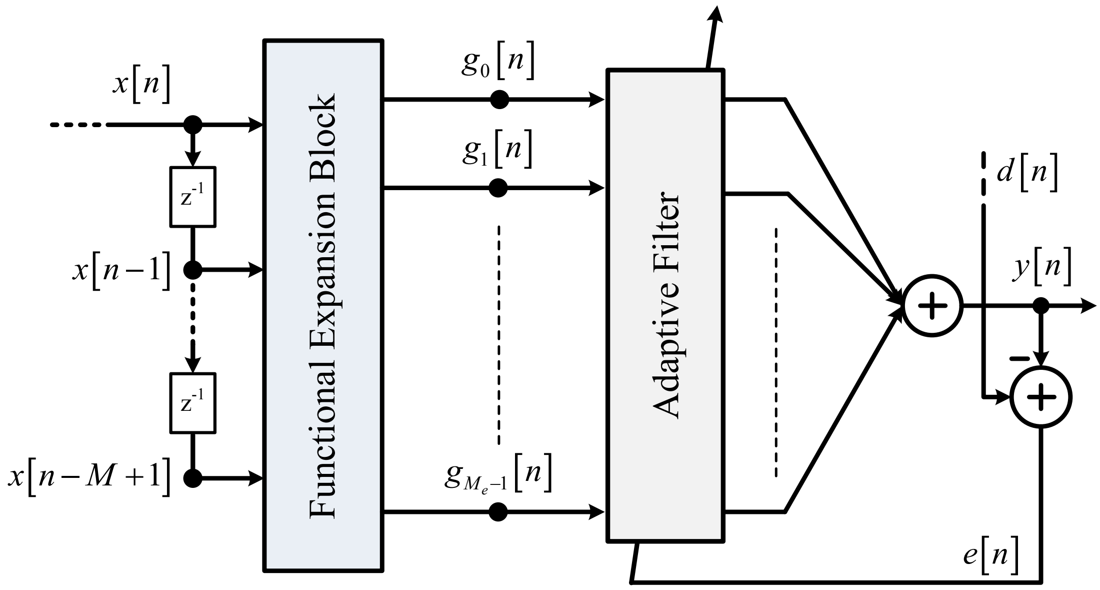

Fig. 8.2: *The functional link adaptive filter.*

the enhanced space are described by the functional links chosen to perform the nonlinear filtering. The functional link adaptive filtering is carried out in two stages: a nonlinear functional expansion of the input and a subsequent linear filtering, as it is possible to see in Fig. 8.2.

At n-th time instant FLAF receives an input buffer $\mathbf{x}_n \in \mathbb{R}^M = \left[x[n] \quad x[n-1] \quad \ldots \quad x[n-M+1]\right]^T$, where M is the input buffer length; differently from the linear weighting carried out by a linear filter, FLAF processes the input buffer by means of a *functional expansion block* (FEB). The FEB generates a series of linearly independent functions, which might be a subset of a complete set of orthonormal basis functions, satisfying universal approximation constraints [34]. The term functional links actually refers to this series of functions. The FEB processes the input buffer by passing each element of the buffer as argument for the chosen functions. The described process results in an *expanded buffer* $\mathbf{g}_n$, whose length is $M_e \geq M$. A deeper description of the expansion process will be drawn in Subsection 8.3.2.

In one sense, no new *ad hoc* information has been inserted into the process; however, the representation of the original buffer has been definitely expanded, and nonlinear modeling becomes possible in the expanded space. Once achieved the expanded buffer, the functional link adaptive filtering process is completed simply linearly filtering the expanded buffer. This aspect is an important theoretical novelty, with respect to the original formulation of functional links [103] and their recent use [162, 125], due to the significant advantages that it provides to FLAF, as described in Subsection 8.3.4.

8.3.2 Nonlinear input expansion

The most important element of the FLAF is the FEB, whose processing plays a leading role in the nonlinear modelling. The expansion process carried out by the FEB is depicted in Fig. 8.3, where it is possible to see how the input buffer $\mathbf{x}_n$ is projected in a higher dimensional space yielding the expanded buffer.

At n-th time instant, the i-th sample of the input buffer $x[n-i]$, being $i = 0, 1, \ldots, M-1$, is expanded by means of a chosen set of functional links $\mathbf{\Phi} = \left\{\varphi_0(\cdot), \varphi_1(\cdot), \ldots, \varphi_{Q-1}(\cdot)\right\}$, where Q is the number of functional links of the chosen set $\mathbf{\Phi}$.

The effectiveness of the FEB relies on two main feature of the chosen set of functional links $\mathbf{\Phi}$. The first feature will be detailed in Section 8.4 and concerns the nature of the expansion and, therefore, the choice of the functional links. The second feature is the correspondence between the input and the output samples of the FEB which can be characterized by the choice of taking into account some memory of the input buffer. This feature will be described in Section 8.5. The former feature depends on the kind of scenario of application and on the nature of involved signals; on the other hand, the latter feature depends on the nature of the input signal and, more specifically, on the kind of distortion which affects the desired signal.

8.3.3 FLAF learning algorithm

Once chosen the set of basis functions, the problem focuses on finding out the coefficients of the FLAF weight vector $\mathbf{w}_n \in \mathbb{R}^{M_e}$, defined as:

$$\mathbf{w}_n = \begin{bmatrix} w_0[n] & w_1[n] & \dots & w_{M_e-1}[n] \end{bmatrix}^T, \tag{8.6}$$

in order to yield the best possible approximation of the nonlinear model within a small error value ε. Therefore, the explicit representation of the FLAF error signal $e[n]$ is:

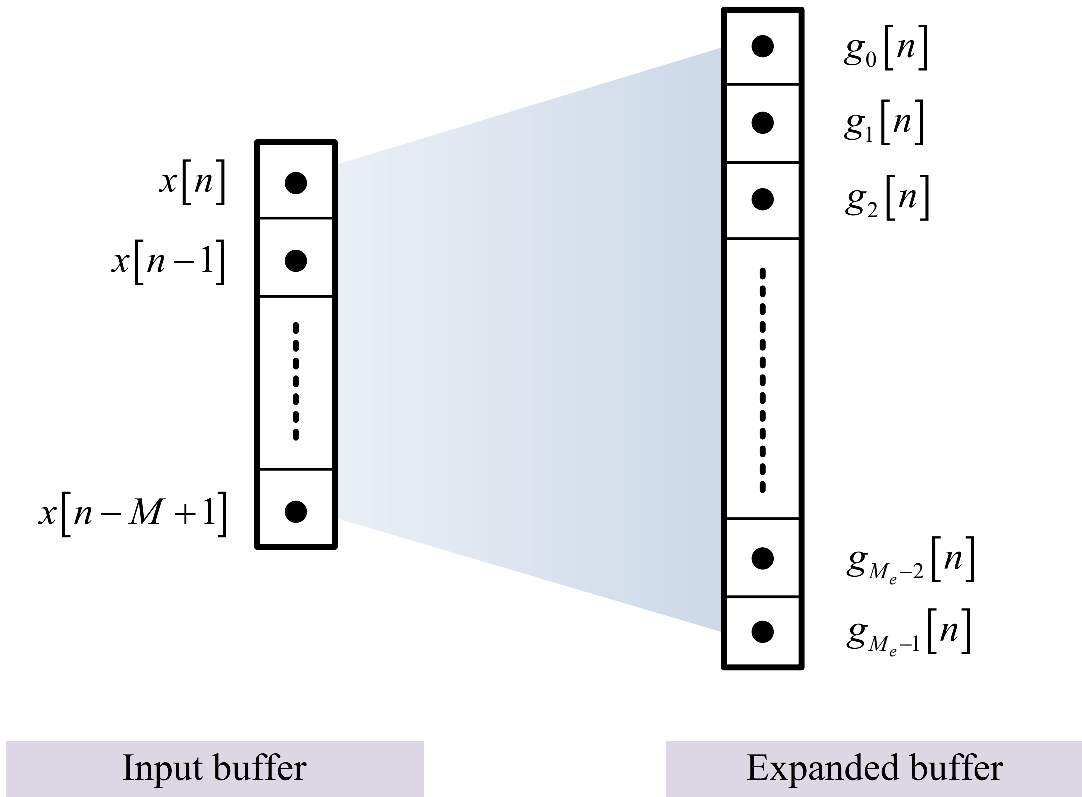

Fig. 8.3: *Functional link expansion.*

$$\begin{aligned} e[n] &= d[n] - y[n] \\ &= d[n] - \mathbf{g}_n^T \mathbf{w}_{n-1} \end{aligned} \tag{8.7}$$

whose minimization depends on a proper estimate of the weights of the filter $\mathbf{w}_n$. In order to find the coefficients of $\mathbf{w}_n$ it is possible to use any adaptive algorithm based on gradient descent rule [120]. In this work we use linear adaptive algorithms based on stochastic gradient rule (see Chapter 4) to adapt the filter coefficients.

8.3.4 Advantages and drawbacks of FLAF

The use of FLAF entails several attractive advantages. Firstly, FLAF has a hugely flexible architecture due to its scalable nonlinear expansion and to its scalable structural complexity. The former property allows to choose *a priori* a suitable series of functional links according to the application of interest. On the other hand, the latter property allows to deal with high dimension input signals, modelling the FEB structure in order to find the right trade-off between performance and computational complexity, according to application requirements and disposable computational resources. Moreover, the flexibility of FLAF architecture allows an easy integration of any *a priori* knowledge of a certain nonlinear system.

Furthermore, it is well known that the introduction of high-order functions in FLAF structure entails an increase of the learning rate [72] and a robust generalizing ability [90]. This property becomes more solid in FLAF, compared to FLANN [109, 72], due to the abilities to exploits the theory of linear adaptive filters [120] by using fast learning algorithms. In addition, the use of a linear filter provides FLAF with significant tracking capabilities that makes it suitable for DSP applications.

However, FLAF might also show some drawbacks, mainly caused by certain applications. A substantial difficulty might be definitely caused by the extreme flexibility of the architecture and in particular by the lack of a well-defined choice of an optimum nonlinear expansion and by a possible need of an *a priori* knowledge of the nonlinear system to design the expansion. Actually FLAF performance is strictly sensitive to the choice of nonlinear functions. Another drawback is that FLAF might incur in a biased convergence resulting in a non-optimum estimation [83, 135]. The described drawbacks will be certainly matter of future researches.

8.4 CHOICE OF FUNCTIONAL EXPANSION TYPE

The functional expansion process can be designed according to models and signals involved in the application. An important choice in the FEB 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 in particular on the signals involved in the processing.

8.4.1 Choosing a proper set of functional links

The FLAF structure is a cascade of a nonlinear expansion and a linear filter; therefore the learning of a FLAF aims at approximating a continuous multivariate function $f(\mathbf{x}_n)$. In FLAF, the approximating function $\hat{f}(\mathbf{x}_n)$ is represented by a set of basis functions and by the coefficients of the adaptive filter $\mathbf{w}_n$. Inside the functional expansion process, a critical point is enacted by the choice of the complete set of orthonormal basis functions and its subset, which represents the functional links actually used. We start to analyze this problem by using a mathematical derivation.

Let I be a compact simply connected subset of $\mathbb{R}^n$ and $\mathcal{L}^m(I)$ be the subset of Lebesgue measurable functions $\hat{f}: I \subset \mathbb{R}^n \to \mathbb{R}^m$ such that the supremum norm of $\hat{f}$, denoted as $\left\|\hat{f}\right\|_I$ is bounded, i.e. $\left\|\hat{f}\right\|_I = \sup_{\mathbf{x}_n \in I}\left|\hat{f}(\mathbf{x}_n)\right| < \infty$. The space of all continuous functions $\hat{f}: I \to \mathbb{R}^m$ is a subset of $\mathcal{L}^m(I)$ and it is denoted as $\mathcal{C}^m(I)$. Let $\mathfrak{B}_Q = \left\{\varphi_j\right\}_{j=0}^{Q}$ be a subset of basis functions of a linearly independent set $\mathfrak{B}_Q \in \mathcal{L}^m(I)$. Being $\hat{f}(\mathbf{x}_n)$ a continuous function over a compact set, according to the Stone-Weierstrass theorem [139], there exist several subsets of $\mathfrak{B}$ that can uniformly approximate $\hat{f}(\mathbf{x}_n)$ by a discriminant:

$$\begin{aligned}\hat{f}(\mathbf{x}_n) &= \sum_{i=0}^{M-1}\sum_{j=0}^{Q-1}\varphi_j(x[n-i])\,w[n-iQ-j-1] \\ &= \mathbf{g}_n^T\mathbf{w}_{n-1}\end{aligned} \tag{8.8}$$

such that:

$$\max_{\mathbf{x}_n \in I}\left|f(\mathbf{x}_n)-\hat{f}(\mathbf{x}_n)\right| < \varepsilon \tag{8.9}$$

where ε is a small threshold, $\mathbf{x}_n \in I \subset \mathbb{R}^n$ is the FLAF input and $\hat{f}(\mathbf{x}_n)$ represents the FLAF output signal, also denoted as $y[n]$.

8.4.2 Most popular functional link sets

The solution of equation (8.8) depends on the existence of the inverse of the correlation matrix of the enhanced buffer. This can be assured by choosing a proper set of basis functions, which have to be linearly independent. Basis functions satisfying this property may be a subset of orthogonal polynomials, like Chebyshev [91], Legendre [107] and trigonometric polynomials [103], or just approximating functions, such as sigmoid [92] and Gaussian functions [22]. In the following we deal with the most employed functional link bases.

Trigonometric basis functions

It has been pointed out that 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 [154]. 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 [109]. Moreover, trigonometric functions are computationally cheaper than power series-based polynomials. Due to its properties, trigonometric polynomial functions are very popular in functional link expansion, ranging from from function approximation applications [103, 72] and channel equalization [162] to active noise control applications [125]. Functional links with trigonometric functions are also used for dynamic system identification [109].

It is possible to generalized the set of functional links using trigonometric basis expansion in the following way:

$$g_j\left[n\right] = \begin{cases} x\left[n-i\right], & j=0 \\ \sin\left(p\pi x\left[n-i\right]\right), & j=2p+1 \\ \cos\left(p\pi x\left[n-i\right]\right), & j=2p+2 \end{cases} \tag{8.10}$$

where $j=0,\ldots,Q-1$ is the functional link index, and $p=0,\ldots,P-1$ is the expansion index, being P the *expansion order*. In (8.10) it is possible to notice that the first element of the set of functional links, $\varphi_0\left(x\left[n-i\right]\right)$, is the replica of the current i-th input sample. In this way, the expanded buffer contains both linear and nonlinear elements.

Chebyshev polynomial functions

It is well known that Chebyshev polynomial functions are endowed with powerful nonlinear approximation capability [76]. This is the reason why their use is widespread in different fields of application. In particular, Chebyshev polynomials have been widely used both in pattern classification [91] and in functional approximation [76] problems. These works pointed out that an ANN with Chebyshev polynomial expansion has universal approximation capability and faster convergence than a MLP network. Moreover, Chebyshev polynomials were also used in FLANN structure [108] for the problem of identification of nonlinear dynamic systems in presence of input plant noise, showing a strong effectiveness. Furthermore, FLANN using Chebyshev expansion has been used in channel equalization [151, 161].

The effectiveness of Chebyshev polynomials is mainly due to the fact that the Chebyshev expansion of an input entry includes functions of the previous functions. 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 increases rapidly [35]. With reference to other power series of the same degree, Chebyshev polynomials are quite computationally cheap and more efficient [76], and this is the reason why they are frequently used for function approximation. However, when the power series converges slowly the computational cost dramatically increases.

Chebyshev functions are easier to compute with respect to trigonometric polynomial functions. Taking into account the i-th input sample $x[n-i]$, the Chebyshev polynomial expansion can be written as:

$$g_j[n] = \begin{cases} 1, & j = 0 \\ x[n-i], & j = 1 \\ 2x^2[n-i] - 1, & j = 2 \\ 2x[n-i]\, g_{j-1}[n] - g_{j-2}[n], & j = 3, \ldots, Q-1 \end{cases} \tag{8.11}$$

in which both linear and nonlinear terms are included, similar to the trigonometric case (8.10).

Legendre polynomial functions

Similar to Chebyshev polynomials, the Legendre functional links provides computational advantage while promising better performance [107]. Legendre polynomial functions have been widely used for function approximation by means of orthonormal ANN [159] and also functional link based ANN [111, 107]. Legendre-based *quadrature amplitude modulation* (QAM) equalizer

[107] performs better than Radial Basis Function (RBF)-based and linear FIR-based equalizers; however, its performance is similar to that of Chebyshev-based equalizer [110].

Considering the i-th input sample $x[n-i]$, the Legendre polynomials are given by:

$$g_j[n] = \begin{cases} 1, & j=0 \\ x[n-i], & j=1 \\ \left(3x^2[n-i]-1\right)/2, & j=2 \\ \left\{(2j-1)\,x[n-i]\,g_{j-1}[n]-(j-1)\,g_{j-2}[n]\right\}/j, & j=3,\ldots,Q-1 \end{cases} \tag{8.12}$$

where, as the previous two cases, both linear and nonlinear elements are involved.

8.5 MEMORY AND MEMORYLESS FLAF

In addition to the choice of considering the type of functional link set, another important choice in the FLAF design concerns the memory of the input buffer, which bears on the correspondence between samples of the input buffer and those of the expanded buffer. The choice of taking into account some memory is strictly related to the nature of the input signal. In particular, it depends a lot on the type of nonlinearity which deteriorates the input signal, in particular on whether the nonlinearity is *instantaneous*, i.e. it is independent from the time instant, or *dynamic*, i.e. the nonlinearity depends even on the time instant.

8.5.1 Memoryless functional links

The simplest and most commonly implemented type of nonlinearity is the *memoryless* (or instantaneous) one. Given an input signal $x[n]$, the generic output of any memoryless nonlinearity can be written as:

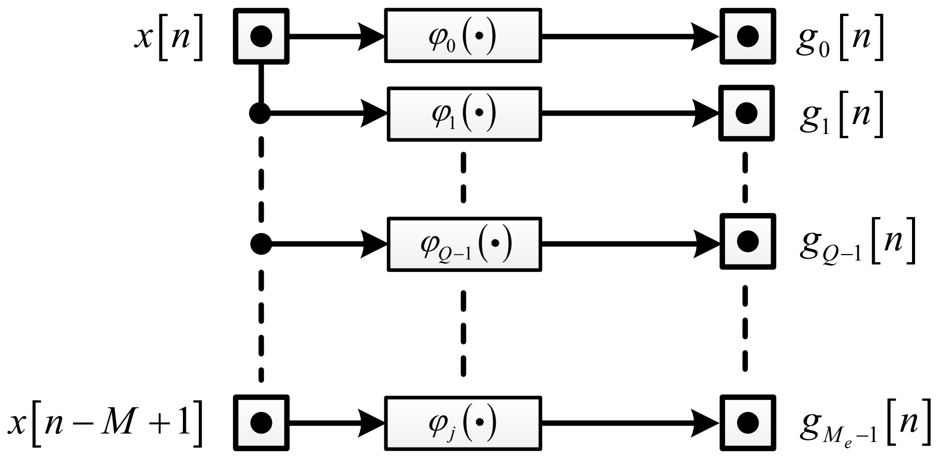

Fig. 8.4: *Functional expansion in memoryless FLAF.*

$$y[n] = f(x[n]) \tag{8.13}$$

where $f(\cdot)$ is some function which maps each input value to a unique output value [127]. Memoryless nonlinearities are very popular since many complex nonlinear systems can be broken down into a linear system containing a memoryless nonlinearity. Memoryless nonlinearities require *memoryless FLAF* which generates an unambiguous relation between the input buffer and the expanded buffer, as depicted in Fig. 8.4.

In a memoryless FLAF, it is possible to define a set $\mathbf{\Phi}_{\text{ml}}$ of memoryless functional links, each of which takes one input sample as argument, yielding the corresponding sample of the expanded buffer. Since the memoryless set is defined as in Subsection 8.3.2, we omit any subscript and refer to it simply as $\mathbf{\Phi}$. For the first $M-1$ input samples we apply the full set of memoryless functional links $\mathbf{\Phi} = \{\varphi_0(\cdot), \varphi_1(\cdot), \ldots, \varphi_{Q-1}(\cdot)\}$; however, for the M-th input sample, we may choose to stop at j-th functional link, with $j = 0, \ldots, Q-1$, or to apply the full set $\mathbf{\Phi}$, depending on whether we want to control the expanded buffer length M_e or not.

8.5.2 Functional links with memory

The set of memoryless functional links described above provides a satisfying approximation of a continuous multivariate function, whether the nonlinearity is instantaneous or dynamic. However, in case of nonlinear dynamic systems, better results may be achieved exploiting the flexibility of the

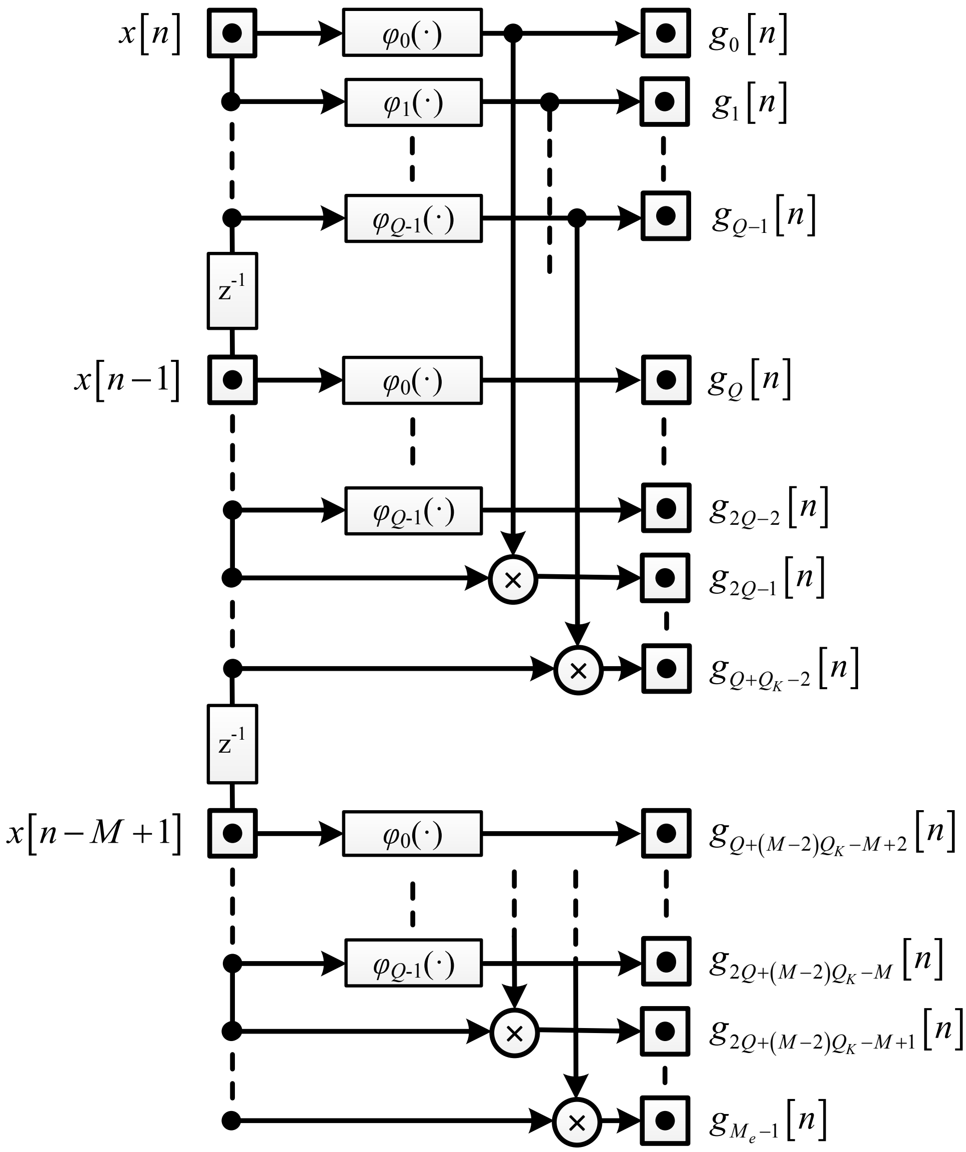

Fig. 8.5: *Functional expansion in FLAF with memory.*

FEB; in particular, it is possible to add to memoryless ones further functional links which take into account the memory of a certain dynamic nonlinearity. We refer to the new set $\mathbf{\Phi}_{\mathrm{m}} = \left\{\varphi_0\left(\cdot\right), \ldots, \varphi_{Q-1}\left(\cdot\right), \varphi_Q\left(\cdot\right), \ldots, \varphi_{Q_K-1}\left(\cdot\right)\right\}$ as a set of *functional links with memory*, where $Q_K > Q$ is the number of functional links with memory. A way of considering the memory of a nonlinearity is that of taking into account the outer products of the *i*-th input sample with the functional links of the previous input samples, as depicted in Fig. 8.5.

In designing the FLAF with memory, it is possible to define a memory order K which 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. Fig. 8.5 shows an expansion with memory order $K = 1$.

8.6 MEAN-SQUARE PERFORMANCE ANALYSIS

8.6.1 Energetic approach to performance analysis

Transient and steady-state performance analyses of adaptive algorithms may be derived considering the expectation and the mean-square of the solution of its stochastic difference equation, which can be described by the expression (8.4). In particular, such analyses are conducted considering the asymptotic solution of the stochastic difference equation, defined as the limit, for $n \to \infty$, of $\mathbf{w}_n$. However, the presence of nonlinearities makes this approach impracticable. An alternative approach for the study of transient and steady-state performance analyses of adaptive algorithms is based on an *energy conservation relation* [120].

We start the derivation considering an important consequence of the data analysis model depicted in Fig. 8.1. Indeed, due to the independence property of the additive noise signal [120], it is possible to neglect $v\left[n\right]$, thus equation (8.3) turns into:

$$d[n] = \mathbf{u}_n^T \mathbf{w}^{\mathrm{opt}} \tag{8.14}$$

Therefore, similarly to equation (8.15), it is possible to define the *a priori* estimation error as:

$$e_a[n] = \mathbf{u}_n^T \mathbf{w}^{\mathrm{opt}} - \mathbf{g}_n^T \mathbf{w}_{n-1}. \tag{8.15}$$

which measures how close the nonlinear estimator $\mathbf{g}_n^T \mathbf{w}_{n-1}$ is to the desired response $d[n]$. Similarly, it is possible to the define the *a posteriori* estimation error as:

$$e_p[n] = \mathbf{u}_n^T \mathbf{w}^{\mathrm{opt}} - \mathbf{g}_n^T \mathbf{w}_n. \tag{8.16}$$

We consider the generic form (8.4) of the Hammerstein nonlinear adaptive filter; multiplying both sides of (8.4) by $\mathbf{g}_n^T$ from the left we obtain:

$$\mathbf{g}_n^T \mathbf{w}_n = \mathbf{g}_n^T \mathbf{w}_{n-1} - \mu \left\|\mathbf{g}_n\right\|^2 \gamma\left(e[n]\right) \tag{8.17}$$

Then, subtracting (8.17) from the desired response defined in (8.14), we achieve a relation between the *a priori* and *a posteriori* error signals:

$$e_p[n] = e_a[n] - \mu \left\|\mathbf{g}_n\right\|^2 \gamma\left(e[n]\right) \tag{8.18}$$

Equation (8.18) provides an alternative description of the stochastic equation (8.4). Generally, it is possible to analyse the behaviour of an adaptive filter in terms of estimation errors, $e_a[n]$ and $e_p[n]$, and in terms of misalignment vector $\widetilde{\mathbf{w}}_n = \mathbf{w}_n - \mathbf{w}^{\mathrm{opt}}$. However, in case of Hammerstein nonlinear filter it is not possible to take into account the information about the misalignment vector, thus the estimation errors are the only useful quantities in order to determine the behaviour of the filter. This is the reason why equation (8.18) assumes a significant relevance, since it turns out to be the only relation from which it is possible to accomplish a performance analysis. In particular, it is possible to derive the following behaviours:

- *Steady-state behaviour*, by means of the expectations $\mathrm{E}\left\{|e_a[n]|^2\right\}$ and $\mathrm{E}\left\{|e[n]|^2\right\}$.
- *Stability*, by determining the range of values of the step-size μ over which $\mathrm{E}\left\{|e_a[n]|^2\right\}$ remains bounded.
- *Transient behaviour*, by studying the evolution of the curve $\mathrm{E}\left\{|e_a[n]|^2\right\}$.

Therefore, in order to address these behaviours we may deal with an energy equality that relates the squared norms of the estimation errors.

8.6.2 Derivation of the energy conservation principle

The energy conservation relation does not depend on the error nonlinearity $\gamma(\cdot)$ [120], thus, in order to generalize this approach, it is possible to use equations (8.18) and (8.4) to solve for $\gamma(\cdot)$, distinguishing between three different cases.

1. $\mathbf{x}_n = \mathbf{0}$.
   The *degenerate case* is common for any linear adaptive filter and both Wiener and Hammerstein-based nonlinear filter. $\mathbf{x}_n = \mathbf{0}$ implies that $\mathbf{u}_n = \mathbf{g}_n = \mathbf{0}$, therefore it is obvious from (8.4) and (8.18) that $\mathbf{w}_n = \mathbf{w}_{n-1}$ and $e_p[n] = e_a[n]$, thus resulting:

$$\|\mathbf{w}_n\|^2 = \|\mathbf{w}_{n-1}\|^2 \quad \text{and} \quad |e_p[n]|^2 = |e_a[n]|^2 \tag{8.19}$$

2. $\mathbf{x}_n \neq \mathbf{0}$, $\mathbf{g}_n = \mathbf{u}_n$.
   As the previous case, this condition is still common for any linear and nonlinear adaptive filter. We solve for $\gamma(\cdot)$ from (8.18), using the constraint $\mathbf{g}_n = \mathbf{u}_n$, and substitute it into (8.4), obtaining:

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \frac{\mathbf{u}_n}{\|\mathbf{u}_n\|^2} \left(e_a[n] - e_p[n]\right) \tag{8.20}$$

It is possible to notice that in equation (8.20) even the step-size μ is cancelled out. Moreover, in equation (8.20) the two estimation errors appear. In order to have an equality between the two errors, it is possible to rearrange equation (8.20):

$$\mathbf{w}_n + \frac{\mathbf{u}_n}{\|\mathbf{u}_n\|^2} e_a[n] = \mathbf{w}_{n-1} + \frac{\mathbf{u}_n}{\|\mathbf{u}_n\|^2} e_p[n]\,. \tag{8.21}$$

If we evaluate the energy of both sides of (8.21), we find out the following energy equality:

$$\|\mathbf{w}_n\|^2 + \frac{1}{\|\mathbf{u}_n\|^2} |e_a[n]|^2 = \|\mathbf{w}_{n-1}\|^2 + \frac{1}{\|\mathbf{u}_n\|^2} |e_p[n]|^2\,. \tag{8.22}$$

in which we do not take into account irrelevant cross-terms in order to have a fair energy relation.

3. $\mathbf{x}_n \neq \mathbf{0}$, $\mathbf{g}_n \neq \mathbf{u}_n$.
The third case is not common for any adaptive filter, but it is specific to a Hammerstein nonlinear adaptive filter. Similarly to case 2 but without using any constraint, we solve for $\gamma(\cdot)$ from (8.18):

$$\gamma(e[n]) = \frac{1}{\mu \|\mathbf{g}_n\|^2} \left(e_a[n] - e_p[n]\right) \tag{8.23}$$

and then we substitute $\gamma(e[n])$ into (8.4), obtaining:

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \frac{\mathbf{g}_n}{\|\mathbf{g}_n\|^2} \left(e_a[n] - e_p[n]\right) \tag{8.24}$$

and the correspondent energy relation:

$$\|\mathbf{w}_n\|^2 + \frac{1}{\|\mathbf{g}_n\|^2} |e_a[n]|^2 = \|\mathbf{w}_{n-1}\|^2 + \frac{1}{\|\mathbf{g}_n\|^2} |e_p[n]|^2 . \tag{8.25}$$

The results achieved in the three different cases can be combined together by defining a common term $\overline{\mu}[n]$:

$$\overline{\mu}[n] = \begin{cases} 0, & \mathbf{x}_n = \mathbf{0} \\ 1/\|\mathbf{u}_n\|^2, & \mathbf{x}_n \neq \mathbf{0}, \quad \mathbf{g}_n = \mathbf{u}_n \\ 1/\|\mathbf{g}_n\|^2, & \mathbf{x}_n \neq \mathbf{0}, \quad \mathbf{g}_n \neq \mathbf{u}_n \end{cases} \tag{8.26}$$

Using (8.26), we can combine (8.19), (8.22) and (8.25) into a single identity:

$$\|\mathbf{w}_n\|^2 + \overline{\mu}[n] |e_a[n]|^2 = \|\mathbf{w}_{n-1}\|^2 + \overline{\mu}[n] |e_p[n]|^2 \tag{8.27}$$

which generalizes the energy conservation relation and provides a unifying framework for the performance analysis of any linear and nonlinear adaptive filters.

Theorem 1 ***Energy conservation relation.*** *For any linear adaptive filter and for both Wiener and Hammerstein model-based nonlinear filter, it always holds that:*

$$\|\mathbf{w}_n\|^2 + \overline{\mu}[n] |e_a[n]|^2 = \|\mathbf{w}_{n-1}\|^2 + \overline{\mu}[n] |e_p[n]|^2$$

where $e_a[n] = \mathbf{u}_n^T \mathbf{w}^{\text{opt}} - \mathbf{g}_n^T \mathbf{w}_{n-1}$, $e_p[n] = \mathbf{u}_n^T \mathbf{w}^{\text{opt}} - \mathbf{g}_n^T \mathbf{w}_n$, *and* $\overline{\mu}[n]$ *is defined as in* (8.26).