Signal Processing 93 (2013) 772–783

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Nonlinear spline adaptive filtering

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

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

ARTICLE INFO

Article history:
Received 23 May 2012
Received in revised form
12 September 2012
Accepted 27 September 2012
Available online 11 October 2012

Keywords:
Spline adaptive filter
Nonlinear adaptive filter
Nonlinear FIR filter
Wiener system identification

ABSTRACT

In this paper a new class of nonlinear adaptive filters, consisting of a linear combiner followed by a flexible memory-less function, is presented. The nonlinear function involved in the adaptation process is based on a spline function that can be modified during learning. The spline control points are adaptively changed using gradient-based techniques. B-splines and Catmull-Rom splines are used, because they allow to impose simple constraints on control parameters. This new kind of adaptive function is then applied to the output of a linear adaptive filter and it is used for the identification of Wiener-type nonlinear systems. In addition, we derive a simple form of the adaptation algorithm and an upper bound on the choice of the step-size. Some experimental results are also presented to demonstrate the effectiveness of the proposed method.

1. Introduction

In recent decades, great attention in the scientific community has been directed to the problem of modeling and identification of nonlinear systems. In the signal processing field linear filters have been employed as a general tool for which a well established theory is available [1,2]. However, in many practical situations linear filters result to be inadequate for system modeling, yielding approximate models that are acceptable only in proximity of a specific operating point [3,1]. Most of the real dynamical systems in extended operating ranges are better described by nonlinear models. Unfortunately, differently from the linear case, a general theoretic framework is not available for the nonlinear one. For this reason nonlinear filtering can be considered more an art than an exact science.

Generally speaking, according to the available *a priori* information, identification problems can be classified as *white-box* or *black-box* problems. White-box or exact structural models are based on availability of complete *a priori* information. For example, in linear adaptive filtering (LAF) the filter order, that indicates the memory length in terms of delay-line discrete-time elements, is the only *a priori* information needed for an exact white-box model definition [3,4].

In contrast, black-box or behavioral models have to be considered when complete information is not available. In this case the linearity assumption is very strong and, even when the exact order is unknown, the LAF basic structure can be considered more as a gray-box rather than a black-box [5]. In contrast, in the nonlinear case an exact structural model (white- or gray-box) is used only in those situations where a consistent mathematical relationship is known. However, these cases are quite unusual and in practice black-box models are almost always employed [3,4].

In order to model nonlinear systems, truncated Volterra series were introduced [6,7]. Volterra series, one of the most used black-box models, are a generalization of the Taylor series expansion based on the convolutive

* Corresponding author. Tel.: +39 06 44585869; fax: +39 06 4873300.
E-mail addresses: michele.scarpiniti@uniroma1.it (M. Scarpiniti), danilo.comminiello@uniroma1.it (D. Comminiello), raffaele.parisi@uniroma1.it (R. Parisi), aurel@ieee.org (A. Uncini).

0165-1684/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2012.09.021

kernel functionals. Due to the large number of free parameters required, the truncated Volterra adaptive filter (VAF) is generally used only in situations of mild nonlinearity [8–11].

In practice, in nonlinear filtering one of the most used structures is the so-called block-oriented representation, in which linear time invariant (LTI) models are connected with memoryless nonlinear functions. The basic classes of block-oriented nonlinear systems are represented by the Wiener and Hammerstein models [3] and by those system architectures originated by the connection of these two classes according to different topologies (i.e. parallel, cascade, feedback etc). The Wiener model consists of a cascade of a linear LTI filter followed by a static nonlinear function and sometimes is known as linear–nonlinear (LN) model [12–14]. The Hammerstein model, which consists of a cascade connection of a static nonlinear function followed by a LTI filter [15,16], is sometimes indicated as nonlinear–linear (NL) model. In addition, sandwich models, such as the linear–nonlinear–linear (LNL) or the nonlinear–linear–nonlinear (NLN) models, were also introduced [17,1]. There exist other approaches based on the block-oriented representation. A good survey of these nonlinear architectures and the methodologies of their identification without using any *a priori* information, can be found in [18].

A general framework for nonlinear filtering is represented by neural networks (NNs) [19]. As a matter of fact, NNs represent an easy and flexible way to realize nonlinear filters. A drawback of this approach is the high computational cost generally required and the difficulties in on-line applications.

In recent years, new efficient models for nonlinear filtering were proposed. For example, derived from the Pao's functional-link networks (FLNs) [20] used for pattern recognition, in [21] a generalized functional-link network (GFLN) has been introduced. These structures are composed by a functional expansion of the input signals (included the cross-term), followed by a linear combiner in order to generate a single output. The GFLNs are characterized by a low computational cost when compared to Volterra filters and are able to guarantee a very good performance. These particular nonlinear networks have been recently and successfully used in nonlinear active noise control [21] and in nonlinear echo cancellation [22].

Nonlinear adaptive filters called kernel adaptive filters (KAF), closely related to the radial basis function (RBF) and regularization networks [19], have been recently introduced [23,24]. The KAF architectures, that represent LAF generalization in reproducing kernel Hilbert spaces, offer the advantage of a simple implementation [25], but have the problem of a continuously increasing network growth [26].

Both KAF and GFLN can be seen as NL block-oriented Hammerstein models and represent a recent and powerful innovation in the field of nonlinear adaptive filtering.

In this paper we present a new class of LN block-oriented Wiener models, called spline adaptive filters (SAFs), belonging to the class of causal shift-invariant recursive nonlinear filters (like VAF, GFLN and KAF). The proposed architecture, differently from recent existing approaches in which some particular and fixed nonlinearities must be chosen [27], is composed by a linear combiner followed by an adaptable look-up table (LUT) addressed by the linear combiner output and interpolated by a local low order polynomial spline curve (Fig. 1). Both the weights of the linear filter and the interpolating points of the LUT can be adapted by minimization of a specified cost function. In particular, in the present work we want principally focus on some salient aspects related to SAF's on-line adaptation, computational analysis and convergence properties.

This paper is organized as described in the following. Section 2 introduces the fundamentals of interpolation methods that are based on spline functions. Section 3 describes the proposed novel Spline Adaptive Filter (SAF) architecture and derives the learning rules for the adaptation of free parameters. Section 4 presents a theoretical proof of the convergence properties of the SAF architecture and Section 5 shows some experimental results. Finally Section 6 concludes the work.

1.1. Notation used in the paper

In this paper matrices are represented by boldface capital letters, i.e. $\mathbf{A} \in \mathbb{R}^{M\times N}$. All vectors are column vectors, denoted by boldface lowercase letters, like $\mathbf{w} \in \mathbb{R}^M = [w[0], w[1], \ldots, w[M-1]]^T$, where $w[i]$ denotes the i-th individual entry of $\mathbf{w}$. In recursive algorithms definition, discrete-time subscript index n is added. For example, the weights vector, calculated according to some

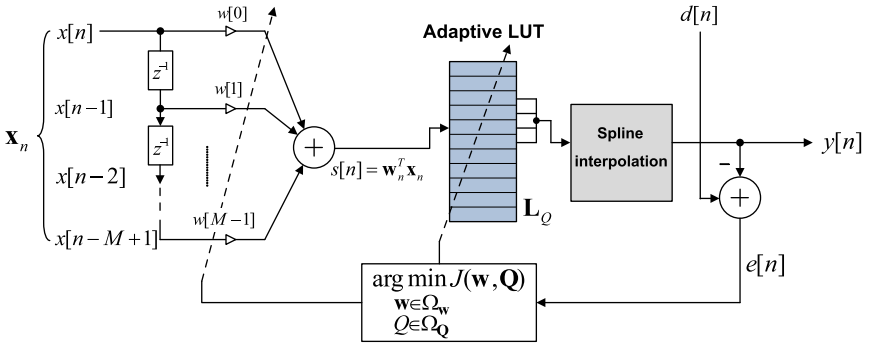

Fig. 1. Structure of a nonlinear spline adaptive filter (SAF).

law, can be written as $\mathbf{w}_{n+1}=\mathbf{w}_n+\Delta\mathbf{w}_n$. In the case of signal regression, vectors are indicated as $\mathbf{x}_n\in\mathbb{R}^M=[x[n], x[n-1],\ldots,x[n-M+1]]^T$, or $\mathbf{x}_{n-1}\in\mathbb{R}^M=[x[n-1],x[n-2],\ldots, x[n-M]]^T$. Note that in the absence of temporal index $\mathbf{x}\equiv\mathbf{x}_n$ by default.

2. Brief review on spline interpolation scheme

Historically derived from elastic rulers that crosses a certain number of points, called *knots*, and used in the past for technical drawing, *splines* are differentiable, up to a given order, polynomial curves for interpolation or approximation of a succession of $Q+1$ knots inside a convex-hull. The knots $\mathbf{Q}_i=[q_{x,i}q_{y,i}]^T$, for $i=0,1,\ldots,Q$ represent a tabulated function on the plane x–y [28–30] and, in order to construct an appropriate function, they are also constrained as

$$q_{x,0}<q_{x,1}<\cdots<q_{x,Q}. \tag{1}$$

Literature on splines is vast. Without loss of generality, in this brief review we consider the univariate basis or blending functions developed by Schoenberg, who introduced the name B-splines, deriving them from Bézier splines [28].

Let $u\in[q_{x,i},q_{x,i+1}]$ be the abscissa value between two consecutive knots. The spline curve is defined as an affine combination of some knots of a suitable P degree spline basis function $N_i^P(u)$ with minimal support and certain continuity properties [30].

$$\varphi(u)=\sum_{i=0}^{Q-P-1}\mathbf{Q}_iN_i^P(u),\quad u\in[q_{x,n},q_{x,Q-P}]. \tag{2}$$

Given $Q+1$ knots, the univariate spline basis functions $N_i^P(u)$ can be defined by the Cox–deBoor recursion [29]

$$N_i^0(u)=\begin{cases}1, & q_{x,i}\le u<q_{x,i+1},\\ 0, & \text{otherwise},\end{cases}$$

$$N_i^P(u)=\frac{u-q_{x,i}}{q_{x,i+P}-q_{x,i}}N_i^{P-1}(u)+\frac{q_{x,i+P+1}-u}{q_{x,i+P+1}-q_{x,i+1}}N_{i+1}^{P-1}(u), \tag{3}$$

where $i=0,1,\ldots,Q-P-1$. The blending functions $N_i^P(u)$ are polynomials of degree P with continuity of the first $P-1$ derivatives. For example, for $P=1$, we have

$$N_i^1(u)=\begin{cases}\dfrac{u-q_{x,i}}{q_{x,i+1}-q_{x,i}}, & q_{x,i}\le u<q_{x,i+1},\\ \dfrac{q_{x,i+2}-u}{q_{x,i+2}-q_{x,i+1}}, & q_{x,i+1}\le u<q_{x,i+2},\\ 0, & \text{otherwise},\end{cases}$$

which results in a classical linear interpolation scheme. So, $N_i^0(u)$ is a rectangular function, $N_i^1(u)$ is a linear double-ramp function, $N_i^2(u)$ is a quadratic double-ramp function. The generic $N_i^P(u)$ can be evaluated as a sequential convolution of P rectangular pulse functions $N_i^P=N_i^0*\cdots*N_i^0$.

Hence, the curve (2) can be written as

$$\varphi(u)=\varphi_i(u),\quad \forall i=0,\ldots,Q \text{ and } u\in[0,1), \tag{4}$$

where $\varphi_i(u)$, is a P degree local polynomial, called curve *span*, entirely defined from $P+1$ knots, that are sometimes called control points. Moreover, scaling or translating the knot vector does not modify the basis functions, that can be fixed *a priori* for all control points.

From (4), there must be a unique mapping that allows us to calculate the local parameter u, as well as the proper curve span index i, from the global abscissa parameter (that in our case is the output of the linear combiner $s[n]=\mathbf{w}^T\mathbf{x}$). In this way, we can represent any point lying on the curve $\varphi(u)$ as a point belonging to the single $\varphi_i(u)$ curve span.

2.1. Uniform quadratic and cubic spline

Recursive Eq. (3) can be entirely pre-calculated with considerable computational saving, by imposing a uniform distribution knot-vector, i.e. constant sampling step of the abscissa $\Delta x=q_{x,i}-q_{x,i+1}$. A similar approach was successfully used in multilayer feed-forward neural networks [31]. It follows that the local polynomial $\varphi_i(u)$ in (4) can be written in a very simple form as

$$\varphi_i(u)=\mathbf{u}^T\mathbf{C}\mathbf{q}_i, \tag{5}$$

where the matrix $\mathbf{C}\in\mathbb{R}^{(P+1)\times(P+1)}$ is a pre-computed matrix, usually called *spline basis matrix*. The vector $\mathbf{u}$ is defined as $\mathbf{u}\in\mathbb{R}^{(P+1)\times1}=[u^P\ u^{P-1}\ \ldots\ u\ 1]^T$, where u is the normalized abscissa value between two knots. The vector $\mathbf{q}_i$ is referred to the control points, defined by $\mathbf{q}_i\in\mathbb{R}^{(P+1)\times1}=[q_i\ q_{i+1}\ \ldots\ q_{i+P}]^T$, where $q_i\equiv q_{y,i}$. Moreover, it is very simple to obtain the derivative of the polynomial $\varphi_i(u)$. In fact, deriving (5) with respect to u, it holds that

$$\frac{\partial\varphi_i(u)}{\partial u}=\varphi_i'(u)=\dot{\mathbf{u}}^T\mathbf{C}\mathbf{q}_i, \tag{6}$$

where $\dot{\mathbf{u}}\in\mathbb{R}^{(P+1)\times1}=\partial\mathbf{u}/\partial u=[Pu^{P-1}\ (P-1)u^{P-2}\ \ldots\ 1\ 0]^T$. This result could be very useful in LMS-like learning algorithms.

For example, for $P=2$, the pre-calculation of the recursion (3) is

$$N_i^2(u)=\begin{cases}\frac{1}{2}(u-q_i), & q_{x,i}\le u<q_{x,i+1},\\ \frac{1}{2}-(u-q_{i+1})-(u-q_{i+1})^2, & q_{x,i+1}\le u<q_{x,i+2},\\ \frac{1}{2}[1-(u-q_{i+2})]^2, & q_{x,i+2}\le u<q_{x,i+3},\end{cases}$$

that can be reformulated in the matrix form (5) as

$$\varphi_i(u)=[u^2\ \ u\ \ 1]\frac{1}{2}\begin{bmatrix}1&-2&1\\-2&2&0\\1&1&0\end{bmatrix}\begin{bmatrix}q_i\\q_{i+1}\\q_{i+2}\end{bmatrix},$$

for $i=0,1,\ldots,Q$ and $u\in[0,1)$. For $P=3$, we have

$$\varphi_i(u)=[u^3\ u^2\ u\ 1]\mathbf{C}\begin{bmatrix}q_i\\q_{i+1}\\q_{i+2}\\q_{i+3}\end{bmatrix},$$

where the basis matrix $\mathbf{C}$ is defined as

$$\mathbf{C}=\frac{1}{6}\begin{bmatrix}-1&3&-3&1\\3&-6&3&0\\-3&0&3&0\\1&4&1&0\end{bmatrix}.$$

Imposing different constraints to approximation relationship, several spline basis with different properties, can be

evaluated in a similar manner. An example of such a basis is the case of *Catmul-Rom* (CR) spline [32], very important in many applications. The interpolation scheme is the same of (5), and the CR-spline basis matrix **C** has the form

$$\mathbf{C} = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}.$$

Note that CR-splines, initially developed for computer graphics purposes, represent a local interpolating scheme. CR-spline, in fact, specifies a curve that pass through all of the control points, a feature which is not necessarily true for other spline methodologies. Overall, the CR-spline results in a more local approximation with respect to the B-spline.

For both CR- and B-basis the curve has a continuous first derivative. For B-spline the second derivative is also a continuous function, while for CR spline the second derivative is not continuous at the knots. However, there is a regularization property common to most of the polynomial spline basis set (CR- and B-spline included) called *variation diminishing property* [29], which ensures the absence of unwanted oscillations of the curve between two consecutive control points, as well as the exact representation of linear segments.

Note also that the entries of the matrix **C** satisfy an important general property: the sum of all the entries is equal to one. This property is a direct consequence of the fact that the spline basis functions, derived from Bézier curves, are a generalization of the Bernstein polynomials and hence they represent a partition of unit (i.e. the sum of all basis function is one) [29,28].

In the following sections the representation properties of SAF and the advantages of this approach in some adaptive filter applications are analyzed. In particular, we show that among all available spline curves, due to their local interpolation and regularization characteristics, the cubic CR- and B-spline are suitable nonlinearities for implementing a flexible curve with some interesting features.

3. Spline adaptive filter

With reference to Fig. 1, in order to compute the filter output $y[n]$, we have to determine the explicit dependence between the linear combiner output $s[n]$ and $y[n]$. Note that there is a direct link between $s[n]$ and $y[n]$; in fact, $y[n]$ is a function of u and i which depends on $s[n]$. In the simple case of uniform spacing of knots and referring to the top of Fig. 2, we constrain the control point abscissas to be equidistant and, most important, not adaptable. Moreover, for the sake of efficiency, another constraint is imposed on the control points, forcing the sampling interval to be centered on the x-axis origin (see Fig. 3).

In practical terms, the computation procedure for the determination of the span index i and the local parameters u of (5) can be expressed by the following equations [31]

$$u = \frac{s[n]}{\Delta x} - \left\lfloor \frac{s[n]}{\Delta x} \right\rfloor,$$

$$i = \left\lfloor \frac{s[n]}{\Delta x} \right\rfloor + \frac{Q-1}{2}, \tag{7}$$

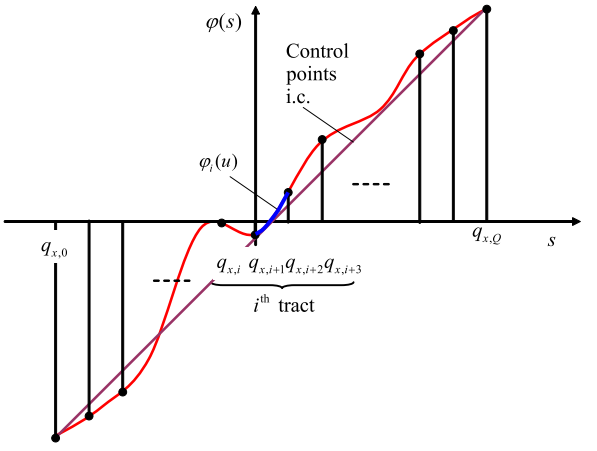

Fig. 3. Example of $q_{y,i}$ control points interpolation using a CR-spline function with a fixed step for the x-axes control points $\Delta x = q_{x,i} - q_{x,i-1}$.

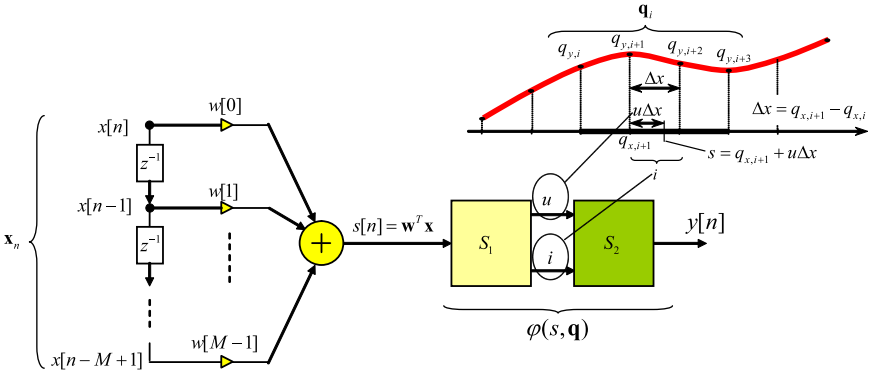

Fig. 2. Schematic structure of the SAF. Block S_1 computes the parameters u an i by (7), while S_2 computes the SAF output through the spline patch determined by S_1.

where $\lfloor\bullet\rfloor$ is the floor operator and the second term in the second equation is an offset value needed to force i to be always nonnegative. Eq. (7), that solves the inversion problem, is inexpensive in terms of computational demand and provide the local variable u and the span index i needed to calculate the output. From the previous discussion and from (5), the output becomes

$$y[n]=\varphi_i(u)=\mathbf{u}^T\mathbf{C}\mathbf{q}_i. \tag{8}$$

We call the two blocks expressed by (7) and (8) S_1 and S_2, respectively (see Fig. 2).

3.1. Adaptation by stochastic-gradient learning rule

With reference to Figs. 1 and 2, in view of (8), the *a priori* error $e[n]$ is defined as

$$e[n]=d[n]-\varphi_i(u). \tag{9}$$

The on-line learning rule can be derived by minimizing a cost function (CF) typically defined as $\hat{J}(\mathbf{w}_n,\mathbf{q}_{i,n})=E\{|e[n]|^2\}$. As usual, this CF can be approximated by the instantaneous error.

$$J(\mathbf{w}_n,\mathbf{q}_{i,n})=e^2[n]. \tag{10}$$

For the minimization of (10), we proceed by applying the stochastic gradient adaptation. Considering the chain rule, at the temporal instant n we can write

$$\frac{\partial J(\mathbf{w}_n,\mathbf{q}_{i,n})}{\partial \mathbf{w}_n}=-2e[n]\frac{\partial\varphi_i(u)}{\partial u}\frac{\partial u}{\partial s[n]}\frac{\partial s[n]}{\partial \mathbf{w}_n}, \tag{11}$$

where $s[n]=\mathbf{w}_n^T\mathbf{x}_n$. From (6) the local derivative of the i-th span is $\partial\varphi_i(u)/\partial u=\varphi_i'(u)=\dot{\mathbf{u}}^T\mathbf{C}\mathbf{q}_{i,n}$ and from expression (7) we have that $\partial u/\partial s[n]=1/\Delta x$. Hence (11) becomes

$$\frac{\partial J(\mathbf{w}_n,\mathbf{q}_{i,n})}{\partial \mathbf{w}_n}=-\frac{2}{\Delta x}e[n]\varphi_i'(u)\mathbf{x}_n. \tag{12}$$

For the derivative computation of (10) with respect to the control points $\mathbf{q}_{i,n}$, we have

$$\frac{\partial J(\mathbf{w}_n,\mathbf{q}_{i,n})}{\partial \mathbf{q}_{i,n}}=\frac{\partial(d[n]-\varphi_i(u))^2}{\partial\varphi_i(u)}\frac{\partial\varphi_i(u)}{\partial \mathbf{q}_{i,n}}, \tag{13}$$

where, from (8), we have that $\partial\varphi_i(u)/\partial\mathbf{q}_{i,n}=\mathbf{C}^T\mathbf{u}$, so we can write

$$\frac{\partial J(\mathbf{w}_n,\mathbf{q}_{i,n})}{\partial \mathbf{q}_{i,n}}=-2e[n]\mathbf{C}^T\mathbf{u}. \tag{14}$$

Finally, indicating explicitly the time index n, the LMS iterative learning algorithm can be written as

$$\mathbf{w}_{n+1}=\mathbf{w}_n+\mu_w e[n]\varphi_i'(u)\mathbf{x}_n, \tag{15}$$

$$\mathbf{q}_{i,n+1}=\mathbf{q}_{i,n}+\mu_q e[n]\mathbf{C}^T\mathbf{u}, \tag{16}$$

where the parameters μ_w and μ_q represent the learning rates for the weights and for the control points, respectively and, for simplicity, incorporate the others constant values. In the next section an analytical derivation of a bound for the learning rates is presented.

It can be noted that at each iteration all the weights are changed, whereas only the four control points of the involved curve span are updated. This is a consequence of the locality of spline interpolation scheme.

A summary of the proposed LMS algorithm for the nonlinear SAF can be found in Algorithm 1.

Algorithm 1. Summary of the SAF-LMS algorithm.

Initialize: $\mathbf{w}_{-1}=\delta[n]$, $\mathbf{q}_0$
1: **for** $n=0,1,\ldots$ **do**
2: $s[n]=\mathbf{w}_n^T\mathbf{x}_n$
3: $u=s[n]/\Delta x-\lfloor s[n]/\Delta x\rfloor$
4: $i=\lfloor s[n]/\Delta x\rfloor+(Q-1)/2$
5: $y[n]=\mathbf{u}^T\mathbf{C}\mathbf{q}_{i,n}$
6: $e[n]=d[n]-y[n]$
7: $\mathbf{w}_{n+1}=\mathbf{w}_n+\mu_w e[n]\dot{\mathbf{u}}^T\mathbf{C}\mathbf{q}_{i,n}\mathbf{x}_n$
8: $\mathbf{q}_{i,n+1}=\mathbf{q}_{i,n}+\mu_q e[n]\mathbf{C}^T\mathbf{u}$
9: **end for**

The learning expressions (15) and (16) can be straightforwardly generalized to second order learning rules like quasi-Newton, Recursive Least Squares (RLS), Affine Projection Algorithm (APA) or other variants [2].

3.2. Computational analysis

From the computational point of view, in addition to the adaptation of the LAF which, for the case of the LMS, is equal to $2M+1$ multiplications plus $2M$ additions for each iteration, we have to consider the adaptation of the control points $\mathbf{q}$.

For each iteration only the i-th span of the curve is modified by calculating the quantities u, i and the expressions $\mathbf{u}^T\mathbf{C}\mathbf{q}_{i,n}$, $\dot{\mathbf{u}}^T\mathbf{C}\mathbf{q}_{i,n}$ and $\mathbf{C}^T\mathbf{u}$ appearing respectively in the 5-th, 7-th and 8-th lines in Algorithm 1. Note that the calculation of the quantity $\mathbf{C}\mathbf{q}$ is executed during the output computation, as well as in the adaptation phase (the spline derivatives). However, most of the computations can be done through the re-use of past calculations (e.g. by explicating the polynomial and by use of well-known Horner's rule [33]). The cost for the spline output computation and its adaptation is $4K_M$ multiplication, plus $4K_A$ additions, where K_M and K_A are constants (less than 16), depending of the implementation structure (see for example [31]). In any case, for high deep memory SAF, where the filter length is $M\gg 4$, the computational overhead, for the nonlinear function computation and its adaptation, can be neglected with respect to a simple linear filter.

4. Convergence properties

The performance of the LMS algorithm is evaluated in terms of its stability behavior, convergence speed, accuracy of the steady-state and transient results and tracking ability as a function of its mean square error (MSE). To achieve optimal performance, it is crucial that the learning rate, used in gradient-based adaptation, is able to adjust in accordance with the dynamics of the input signal $x[n]$ and the nonlinearity $\varphi_i(u)$ [34]. For this purpose, it is useful to adopt an adaptive learning rate, that minimizes the instantaneous output error of the filter [35]. The following analysis is performed in a stationary environment and the learning algorithm is assimilated to a dynamic system described by a stochastic difference equation (SDE).

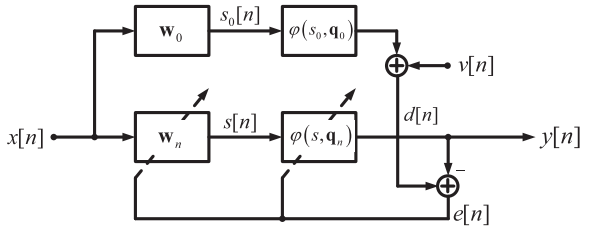

Fig. 4. Adaptive identification scheme used for statistical performance evaluation.

In the case of LAF, the statistical properties are analyzed with respect to an optimal reference solution $(\mathbf{w}_0,\mathbf{q}_0)$ (see Fig. 4) [3]. Defining the error vectors $\mathbf{v}_n^{(\mathbf{w})}=\mathbf{w}_n-\mathbf{w}_0$ and $\mathbf{v}_n^{(\mathbf{q})}=\mathbf{q}_n-\mathbf{q}_0$, in order to demonstrate the weak convergence it is necessary to prove that $\lim_{n\to\infty}E\{\mathbf{v}_n^{(\mathbf{w},\mathbf{q})}\}=\mathbf{0}$. This issue can be achieved by performing a Taylor expansion of the error $e[n]$, that is a nonlinear function of the filter input $x[n]$. The method determines the optimal learning rate in order to assure the convergence.

The cost function (10) depends on two variables. In this sense it is easy to verify that it cannot admit a unique solution so that $\lim_{n\to\infty}E\{\mathbf{w}_n\}=\mathbf{w}_0$ and $\lim_{n\to\infty}E\{\mathbf{q}_n\}=\mathbf{q}_0$, because the variables $\mathbf{w}$ and $\mathbf{q}$ are not independent. In order to demonstrate the convergence property, the adaptation can be performed in two separate phases. For example, the spline control points $\mathbf{q}$ can be preemptively adapted in the first phase of learning. In this way, the hypothesis that the change in control points can be minimal during the last phase of adaptation can be considered true.

The convergence property of the (15) can be derived from the Taylor series expansion of the error $e[n+1]$ around the instant n, stopped at the first order [36,37]

$$e[n+1]=e[n]+\frac{\partial e[n]}{\partial \mathbf{w}_n^T}\Delta\mathbf{w}_n+\text{h.o.t.}, \tag{17}$$

where h.o.t. means high order terms. Now, using (9) and (15), we derive

$$\frac{\partial e[n]}{\partial \mathbf{w}_n^T}=-\varphi_i'(u)\mathbf{x}_n^T, \tag{18}$$

$$\Delta\mathbf{w}_n=\mu_w[n]e[n]\varphi_i'(u)\mathbf{x}_n. \tag{19}$$

Substituting (18) and (19) in (17), we can obtain after simple manipulations

$$e[n+1]=[1-\mu_w[n](\varphi_i'^2(u)\|\mathbf{x}_n\|^2)]e[n]. \tag{20}$$

In order to ensure the convergence of the algorithm, we desire that the norm of the error $e[n+1]$ is not greater than the norm of the right side of (20), assuring the uniform convergence, that is

$$|e[n+1]|\le|1-\mu_w[n](\varphi_i'^2(u)\|\mathbf{x}_n\|^2)|\,|e[n]|. \tag{21}$$

This aim is reached if the following relation holds

$$|1-\mu_w[n](\varphi_i'^2(u)\|\mathbf{x}_n\|^2)|\le 1, \tag{22}$$

that implies the following bound on the choice of the learning rate $\mu_w[n]$

$$0<\mu_w[n]\le\frac{2}{\varphi_i'^2(u)\|\mathbf{x}_n\|^2}. \tag{23}$$

Note that in (23) all quantities are positive, thus no positivity constrains are needed. In addition, the convergence properties depend on the slope, but not on the sign of the derivative. The potential problem of diverging of $\varphi_i'(u)$ is avoided by its intrinsic definition (6). In contrast the division by zero can be easily avoided by introducing a regularization term [2].

In order to stabilize the expression in (23) due to unsatisfactory statistics for nonlinear and nonstationary input signals, an adaptable regularizing term $\delta_w[n]$ can be added. So the learning rate is chosen as

$$\mu_w[n]=\frac{2}{\varphi_i'^2(u)\|\mathbf{x}_n\|^2+\delta_w[n].} \tag{24}$$

The regularizing term $\delta_w[n]$ can be adapted using a gradient descent approach [37,38]:

$$\delta_w[n+1]=\delta_w[n]-\eta_w\nabla_{\delta_w}J(\mathbf{w}_n,\mathbf{q}_{i,n}), \tag{25}$$

where η_w is the learning rate. For the evaluation of the gradient in (25), we use the chain rule

$$\begin{aligned}\nabla_{\delta_w}J(\mathbf{w}_n,\mathbf{q}_{i,n})&=\frac{\partial J(\mathbf{w}_n,\mathbf{q}_{i,n})}{\partial\delta_w[n]}=\frac{\partial J(\mathbf{w}_n,\mathbf{q}_{i,n})}{\partial e[n]}\frac{\partial e[n]}{\partial y[n]}\frac{\partial y[n]}{\partial s[n]}\\&\cdot\frac{\partial s[n]}{\partial\mathbf{w}_n}\frac{\partial\mathbf{w}_n}{\partial\mu[n]}\frac{\partial\mu[n]}{\partial\delta_w[n]}\\&=\frac{\varphi_i'^2(u)e[n]e[n-1]\mathbf{x}_n^T\mathbf{x}_{n-1}}{(\varphi_i'^2(u)\|\mathbf{x}_n\|^2+\delta_w[n-1])^2.}\end{aligned} \tag{26}$$

Hence, we derive

$$\delta_w[n+1]=\delta_w[n]-\eta_w\frac{\varphi_i'^2(u)e[n]e[n-1]\mathbf{x}_n^T\mathbf{x}_{n-1}}{(\varphi_i'^2(u)\|\mathbf{x}_n\|^2+\delta_w[n-1])^2.} \tag{27}$$

It is straightforward to note the advantage of (27) for colored input signal $x[n]$.

In a similar way, we can derive a bound on $\mu_q[n]$. From the Taylor series expansion of the error $e[n+1]$ around the instant n, stopped at the first order, we obtain:

$$e[n+1]=e[n]+\frac{\partial e[n]}{\partial\mathbf{q}_{i,n}^T}\Delta\mathbf{q}_{i,n}+\text{h.o.t.}, \tag{28}$$

and, from (9) and (16), the equations

$$\frac{\partial e[n]}{\partial\mathbf{q}_{i,n}^T}=-\mathbf{u}^T\mathbf{C}, \tag{29}$$

$$\Delta\mathbf{q}_{i,n}=\mu_q[n]e[n]\mathbf{C}^T\mathbf{u}. \tag{30}$$

Hence we derive, after simple manipulations

$$e[n+1]=[1-\mu_q[n]\|\mathbf{C}^T\mathbf{u}\|^2]e[n]. \tag{31}$$

In order to ensure the convergence of the algorithm, imposing the uniform convergence of (31) as done in (21), we obtain

$$|1-\mu_q[n]\|\mathbf{C}^T\mathbf{u}\|^2|\le 1, \tag{32}$$

that implies the following bound on the choice of the learning rate $\mu_q[n]$

$$0<\mu_q[n]\le\frac{2}{\|\mathbf{C}^T\mathbf{u}\|^2}. \tag{33}$$

The denominator of (33) is always different from zero, since the particular structure of $\mathbf{u}$ and $\mathbf{C}$. In this case, since the spline nonlinearity imposes a auto-regularizing

constraint, as deeply detailed in [39], there is not need of any regularizing term.

Eqs. (23) and (33) impose on the learning algorithm a simple constraint. The resulting algorithms belong to a general class of nonlinear normalized adaptive filtering methods [2].

Moreover, it is interesting to underline the relation between the learning rate and the slope of the time-varying nonlinearity. In particular, from (15) and (23) we can note that, referring with the R superscript to a reference architecture with unitary slop $\varphi_i'R(u)=1$, the weights variation and the learning rate can be related as

$$\Delta\mathbf{w}_n^R=\varphi_i'(u)\Delta\mathbf{w}_n,$$
$$\mu_w^R[n]=\varphi_i'^2(u)\mu_w[n]. \tag{34}$$

A such kind of intimate relation was already noted in the contest of neural networks [40,41].

5. Experimental results

In order to validate the proposed nonlinear adaptive filtering solutions, several experiments were performed. Experimental tests address toward nonlinear system identification problem, using uniform cubic spline nonlinearities, that represent a good compromise between flexibility and approximation behavior. Comparisons with a full 3-rd order Volterra architecture were also performed.

During the validation phase of our work, we performed many experimental tests, but for paper length optimization, and in order to focus on the simplicity of the proposed approach, we have decided to present only four experiments.[1]

As explained in Section 4, the cost function may not admit an unique minimum value and thus the convergence toward the optimum cannot be guaranteed. In order to avoid this ambiguity, a particular attention must be posed on the choice of the initial conditions for the filter weights and spline control points. Without any specific *a priori* knowledge, a good choice of initial conditions that have always guaranteed excellent results is $\mathbf{w}_{-1}=\delta[n]$ for filter weights, while spline control points $\mathbf{q}_{-1}$ are initialized as a straight line with a unitary slope.

5.1. Experiment 1

A first experiment is performed in order to demonstrate the convergence behavior of the SAF illustrated in Section 4. The experiment consists in the identification of an unknown Wiener system composed by a linear component $\mathbf{w}_0=[0.6,-0.4,0.25,-0.15,0.1]^T$ and a nonlinear memoryless target function implemented by a 21 points length LUT $\mathbf{q}_0$, interpolated by a uniform third degree spline with an interval sampling $\Delta x=0.2$ defined as

$$\mathbf{q}_0=\{-2,-1.8,\ldots,-1.0,-0.8,-0.91,0.42,-0.01,$$
$$-0.1,0.1,-0.15,0.58,1.2,1.0,1.2,\ldots,2.0\}.$$

[1] Some Matlab source codes, implementing the experimental results, are available at the following web page: http://ispac.ing.uniroma1.it/SAF.html.

The input signal $x[n]$ consists in 20,000 samples of the signal generated by the following relationship

$$x[n]=ax[n-1]+\sqrt{1-a^2}\,\xi[n], \tag{35}$$

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 into the interval [0, 0.99] (see below). In addition it is considered an additive white noise $v[n]$ (see Fig. 4) with a signal to noise ratio $SNR=30$ dB. The learning rates are set to $\mu_w=\mu_q=0.025$, while $\delta_w=0.01$.

Results, averaged over 100 trials, are summarized in Table 1, that shows mean values and variances of each filter tap, while Table 2 shows mean values and variances of each spline control points. We can note that the variances in the case of CR-spline nonlinearity are smaller than in the case of B-spline. This fact is due to the more local behavior of CR-spline. In fact the first scheme imposes the spline to pass through the control points; using B-spline instead, the interpolation is constrained to lie within the convex hull related to the control points [42]. This fact implies that the adaptation of the i-th span modifies only the region between the related knots for the CR-spline, while in the case of B-Spline the adaptation involves the entire curve. In this way a learned profile can be destroyed by a new single value, acting as a forgetting factor on learning procedure for the nonlinearity. This reasoning explains the motivation for higher variances.

In addition Fig. 5 shows a comparison of the MSE for the proposed experimental test with the two different choices of the parameter $a=0.1$ and $a=0.95$. It is straightforward to

Table 1
Mean values and variance of filter taps in experiment 1 averaged over 100 trials, for CR-spline and B-spline, respectively.

i	$\mathbf{w}_0$	CR-spline		B-spline	
		Mean	Variance $\times 10^{-3}$	Mean	Variance $\times 10^{-3}$
1	0.60	0.607	3.342	0.633	15.042
2	−0.40	−0.404	2.434	−0.421	6.951
3	0.25	0.253	1.181	0.264	3.196
4	−0.15	−0.152	1.409	−0.158	1.808
5	0.10	0.101	1.043	0.105	1.287

Table 2
Mean values and variance of spline control points in experiment 1 averaged over 100 trials, for CR-spline and B-spline, respectively.

i	$\mathbf{q}_0$	CR-spline		B-spline	
		Mean	Variance $\times 10^{-3}$	Mean	Variance $\times 10^{-3}$
7	−0.80	−0.812	5.506	−0.838	5.936
8	−0.91	−0.989	5.804	−0.856	29.721
9	−0.42	−0.403	8.460	−0.360	47.493
10	−0.01	−0.008	1.748	−0.015	1.550
11	−0.10	−0.103	1.556	−0.098	1.037
12	0.10	0.102	1.216	0.097	1.245
13	−0.15	−0.152	1.773	−0.138	5.362
14	0.58	0.536	61.013	0.433	3.046
15	1.20	1.182	13.714	1.109	5.547

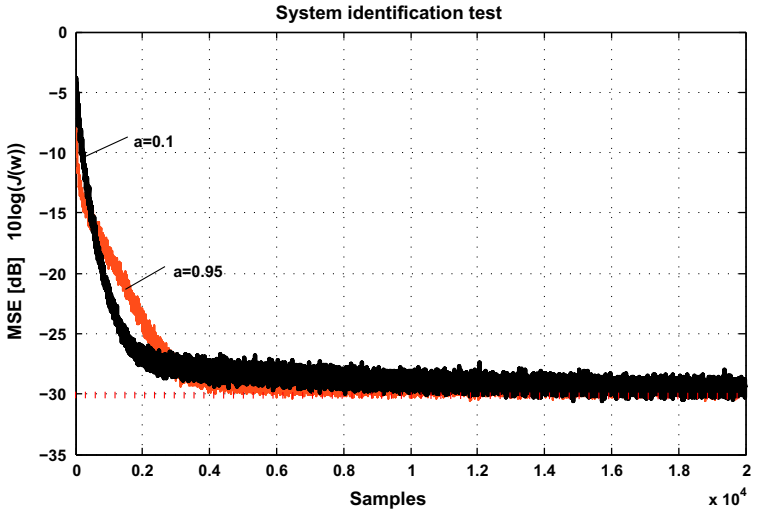

Fig. 5. Comparison of MSE for experiment 1 using model (35) with $a=0.1$ and 0.95, respectively.

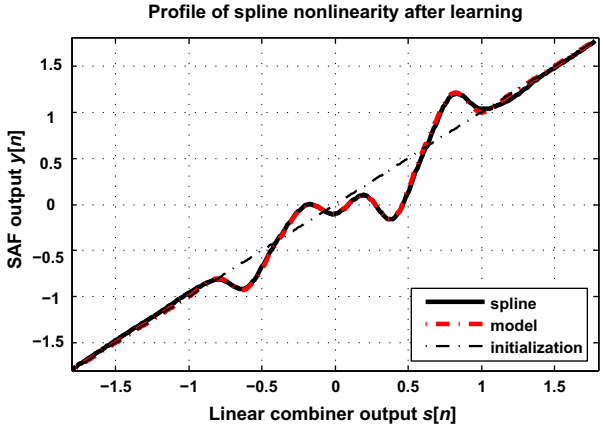

Fig. 6. Comparison of the model and adapted nonlinearity for experiment 1 using model (35) with $a=0.95$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

note that, at steady state, the performance of the SAF algorithm reaches the value of the noise power. Fig. 6 shows the profile of the nonlinearity used in the model (the red dashed line), the profile of the adapted nonlinearity obtained by spline interpolation in SAF (the solid black line) and the initialization of the control points (the dash-dot line). Note that the adapted nonlinearity is perfectly overlapping the model one, as it is also evident from Table 2.

In order to better understand the behavior of the proposed approach by varying the a parameter in (35), Fig. 7 shows the MSE for the following 14 different choices of a

$$a=\{0,\ 0.1,\ \ldots,0.8,\ 0.9,\ 0.95,\ 0.97,\ 0.98,\ 0.99\}.$$

For a better visualization purpose an additional smoothing is also applied to the each MSE graphic. In addition, to enhance the visual comprehension of Fig. 7, the number of iterations for which the MSE reaches -25 dB is forgiven in Table 3, for the last eight values of a.

5.2. Experiment 2

A second experimental set-up is drawn by [43] and consists in the identification of a nonlinear dynamic system composed by three blocks. The first and last blocks are two fourth order IIR filter, Butterworth and Chebychev, respectively, with transfer functions

$$H_B(z)=\frac{(0.2851+0.5704z^{-1}+0.2851z^{-2})}{(1-0.1024z^{-1}+0.4475z^{-2})}\times\frac{(0.2851+0.5701z^{-1}+0.2851z^{-2})}{(1-0.0736z^{-1}+0.0408z^{-2})}, \tag{36}$$

and

$$H_C(z)=\frac{(0.2025+0.2880z^{-1}+0.2025z^{-2})}{(1-1.01z^{-1}+0.5861z^{-2})}\times\frac{(0.2025+0.0034z^{-1}+0.2025z^{-2})}{(1-0.6591z^{-1}+0.1498z^{-2})}, \tag{37}$$

while the second block is the following nonlinearity

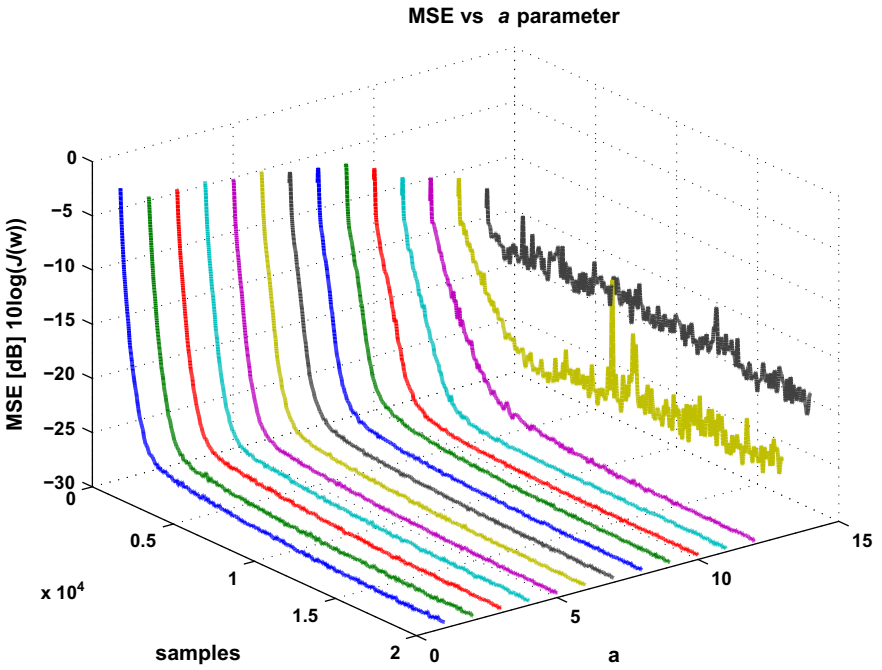

Fig. 7. Comparison of MSE using 14 different values of the a parameter.

Table 3
Number n of iterations for which the MSE reaches -25 dB, in relation to the i-th value of a. Table reports only the last eight values of a, since for the first six the behavior is very similar.

i	a	n	i	a	n
7	0.6	1460	11	0.95	2272
8	0.7	1516	12	0.97	2864
9	0.8	1657	13	0.98	4102
10	0.9	1854	14	0.99	∞

$$y[n] = \frac{2x[n]}{1+|x[n]|^2}. \tag{38}$$

This system is similar to radio frequency amplifiers for satellite communications (high power amplifier), in which the linear filters model the dispersive transmission paths, while the nonlinearity models the amplifier saturation. The input signal $x[n]$ is a zero mean white Gaussian noise with unitary variance. The learning rates are set to $\mu_w = \mu_q = 0.02$, while $\delta_w = 0.01$. We compare the proposed SAF architecture using $M=15$ filter taps with two 3-rd order Volterra filters using $M=5$ and $M=15$ respectively. A comparison of the smoothed MSE over 100 trials for the three architectures is reported in Fig. 8, where is clearly observable the goodness of the proposed approach. It can be noted that the computational complexity of a full third order Volterra filter is very high. In fact, for a filter length M, the complexity is $O(M^3)$ against the simplicity of SAF approach.

5.3. Experiment 3

This experiment consists in the identification of a cubic nonlinear dynamic system described by the following input-output characteristic [44]

$$y[n] = \frac{y[n-1]}{(1+y^2[n-1])} + x^3[n]. \tag{39}$$

where the signal $x[n]$ is an autoregressive process defined as in [45],

$$x[n] = 1.79x[n-1] - 1.85x[n-2] + 1.27x[n-3] - 0.41x[n-4] + \xi[n], \tag{40}$$

and the driving signal $\xi[n]$ is a zero mean white Gaussian noise with unitary variance. The SAF architecture is composed by filter length $M=14$, $\Delta x = 0.05$, while learning rates are set to $\mu_w = \mu_q = 0.5$ and $\delta_w = 0.01$. In order to test the flexibility of the SAF, the simulation are drawn using different input signal normalization conditions [0, 0.1], [−0.1, 0.1] and [−0.25, 0.25]. A comparison of the smoothed MSE over 100 trial is reported in Fig. 9.

Although the proposed architecture is simply feedforward, we can note that SAF filter shows similar optimal performance of the more complex recurrent network in [45] where a similar experiment has been carried out only for [0, 0.1] input signal normalization.

5.4. Experiment 4

In this test we want to prove the robustness of the proposed approach with respect to the Δx parameter, chosen in the set $\Delta x = \{0.15, 0.3, 0.45, 0.6\}$. The system to identify is the Back and Tsoi NARMA model reported in [46]. This model consists in a cascade of the following third order IIR filter

$$H(z) = \frac{0.0154 + 0.0462z^{-1} + 0.0462z^{-2} + 0.0154z^{-3}}{1 - 1.99z^{-1} + 1.572z^{-2} - 0.4583z^{-3}}, \tag{41}$$

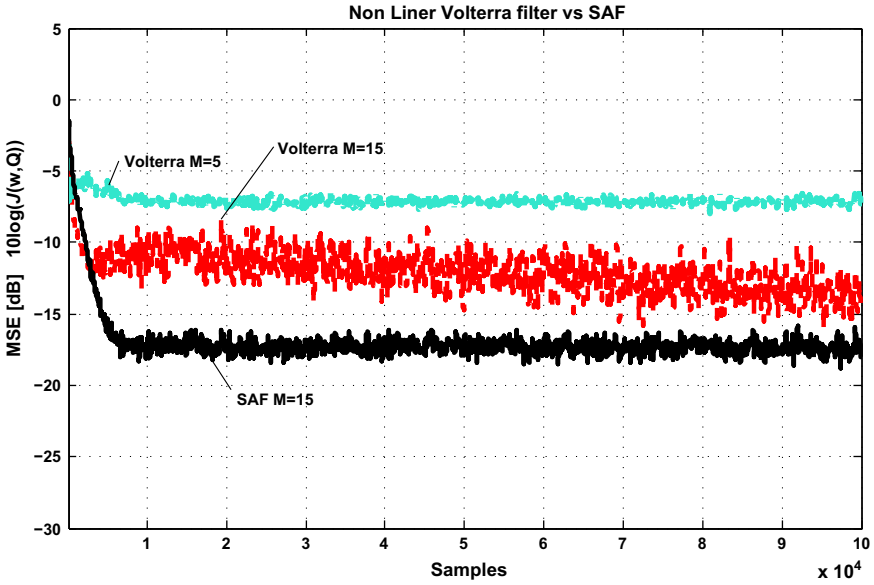

Fig. 8. Comparison of MSE for the third order Volterra filter and the nonlinear SAF for the Panicker–Mathews–Sicuranza system in (36)–(38).

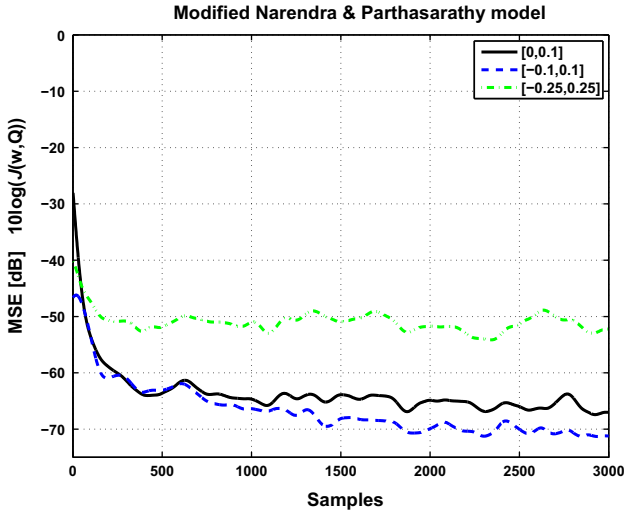

Fig. 9. Comparison MSE for the nonlinear SAF for the modified Narendra and Parthasarathy model in [45] using different normalization conditions.

and the following nonlinearity

$$y[n] = \sin(x[n]). \tag{42}$$

The input signal $x[n]$ is the colored signal obtained with (35), choosing $a=0.95$. The learning rates are set to $\mu_w = \mu_q = 0.02$, $\delta_w = 0.01$ and the filter length is $M=5$ taps, for both the SAF and the third order Volterra filter.

The MSE, averaged over 100 trials, is reported in Fig. 10. Also in this case, these figures clearly show the robustness and the superiority of our approach with respect to Volterra system. This test suggests the possibility of using a greater value for the variable Δx in order to decrease the number of parameters in the LUT collecting the spline control points.

6. Conclusion

This paper introduces a novel nonlinear adaptive filtering model, where the nonlinearity is implemented using spline functions. Splines are flexible nonlinear functions, whose shape can be modified during the learning process using gradient-based techniques. The system implemented in this work is based on Wiener architecture.

The on-line adaptation of the proposed nonlinear spline adaptive filter is based on the well-known LMS algorithm. In addition a constraint on the choice of the learning rate is also derived, in order to assure the algorithm convergence.

Several experimental tests have demonstrated the effectiveness of the proposed approach. In particular SAF

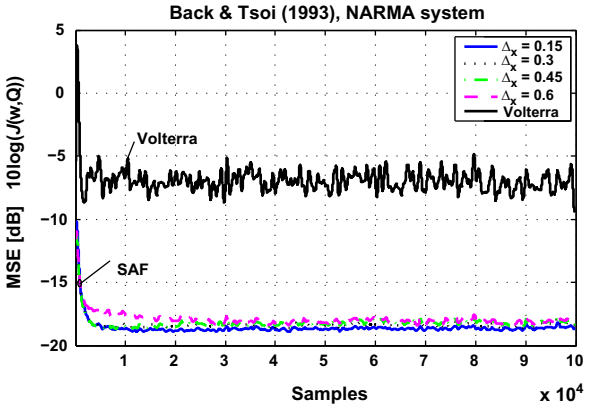

Fig. 10. MSE of the proposed approach for the Back and Tsoi NARMA model in (41) and (42) with different Δx values, compared with a third order Volterra filter.

can reach a good performance with low computational cost, against other existing approaches.

Due to their on-line approach, SAFs can be used in many practical cases. In signal processing typical on-line applications are, for example, interference canceling problems, such as nonlinear acoustic echo cancelers, nonlinear adaptive noise cancelers and nonlinear channel equalization. Another area of interest could be identified in the field of biomedical data capture and analysis. In addition, for the flexibility of the proposed approach, SAFs can be also used in control applications, like vehicular dynamic control or industrial plants control in substitution of the computational expensive Volterra controllers.

Acknowledgment

The authors would like to thank all anonymous reviewers and the editor for their precious and helpful comments and suggestions in improving the quality of this work.

References

[1] L. Ljung, System Identification – Theory for the User, 2nd ed. Upper Saddle River, NJ, 1999.
[2] A.H. Sayed, Fundamentals of Adaptive Filtering, John Wiley & Sons, 2003.
[3] N. Wiener, Nonlinear Problems in Random Theory, John Wiley & Sons, 1958.
[4] O. Montiel, O. Castillo, P. Melin, R. Sepúlveda, Black box evolutionary mathematical modeling applied to linear systems, International Journal of Intelligent Systems 20 (2) (2005) 293–311.
[5] R.K. Pearson, M. Pottmann, Gray-box identification of block-oriented nonlinear models, Journal of Process Control 10 (4) (2000) 301–3115.
[6] V. Volterra, Sopra le funzioni che dipendono da altre funzioni, Rend. Accademia Nazionale dei Lincei (1887) 97–105, 141–146, 15–158.
[7] V. Volterra, Theory of Functionals and of Integral, and Integral-Differential Equations, Blackie & Son Limited, London and Glasgow, 1930.
[8] M. Schetzen, Nonlinear system modeling based on the Wiener theory, Proceedings of the IEEE 69 (12) (1981) 1557–1573.
[9] M. Schetzen, The Volterra and Wiener Theories of Nonlinear Systems, John Wiley & Sons, New York, 1980.
[10] V.J. Mathews, G.L. Sicuranza, Polynomial Signal Processing, John Wiley & Sons, 2000.
[11] J.-M. Le Caillec, Spectral inversion of second order Volterra models based on the blind identification of Wiener models, Signal Processing 91 (11) (2011) 2541–2555.
[12] E.W. Bai, Frequency domain identification of Wiener models, Automatica 39 (2003) 1521–1530.
[13] M. Korenberg, I. Hunter, Two methods for identifying Wiener cascades having noninvertible static nonlinearities, Annals of Biomedical Engineering 27 (6) (1999) 793–804.
[14] D. Westwick, M. Verhaegen, Identifying MIMO Wiener systems using subspace model identification methods, Signal Processing 52 (2) (1996) 235–258.
[15] E.W. Bai, Frequency domain identification of Hammerstein models, IEEE Transactions on Automatic Control 48 (2003) 530–542.
[16] I.J. Umoh, T. Ogunfunmi, An affine projection-based algorithm for identification of nonlinear hammerstein systems, Signal Processing 90 (6) (2010) 2020–2030.
[17] L. Ljung, Identification of nonlinear systems, in: 9th International Conference on Control, Automation, Robotics and Vision (ICARCV 2006), Singapore, 2006.
[18] R. Haber, H. Unbehauen, Structure identification of nonlinear dynamic systems – a survey on input/output approaches, Automatica 26 (4) (1990) 651–677.
[19] S. Haykin, Neural Networks and Learning Machines, 2nd ed. Pearson Publishing, 2009.
[20] Y.H. Pao, Adaptive Pattern Recognition and Neural Networks, Addison-Wesley, 1989.
[21] G.L. Sicuranza, A. Carini, A generalized FLANN filter for nonlinear active noise control, IEEE Transactions on Audio, Speech and Language Processing 19 (8) (2011) 2412–2417.
[22] D. Comminiello, L.A. Azpicueta-Ruiz, M. Scarpiniti, A. Uncini, J. Arenas-García, Functional link based architectures for nonlinear acoustic echo cancellation, in: Proceedings of the Hands-free Speech Communication and Microphone Arrays (HSCMA2011), Edinburgh, UK, 2011, pp. 180–184.
[23] W. Liu, J.C. Principe, S. Haykin, Kernel Adaptive Filtering: A Comprehensive Introduction, Wiley, 2010.
[24] C. Richard, J. Bermudez, P. Honeine, Online prediction of time series data with kernels, IEEE Transactions on Signal Processing 57 (3) (2009) 1058–1067.
[25] W.D. Parreira, J.C.M. Bermudez, C. Richard, J.-Y. Tourneret, Stochastic behavior analysis of the Gaussian kernel least-mean-square algorithm, IEEE Transactions on Signal Processing 60 (5) (2012) 2208–2222.
[26] S. Van Vaerenbergh, I. Santamaria, W. Liu, J.C. Principe, Fixed-budget kernel recursive least-squares, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2010), Dallas, Texas, 2010, pp. 1882–1885.
[27] D. Wang, F. Ding, Least squares based and gradient based iterative identification for Wiener nonlinear systems, Signal Processing 91 (5) (2011) 1182–1189.
[28] L. Schumaker, Spline Functions: Basic Theory, 3rd ed. Krieger Publishing Company, 1993.
[29] C. de Boor, A Practical Guide to Splines, revised ed. Springer-Verlag, 2001.
[30] H. Prautzsch, W. Boehm, M. Paluszny, Bézier and B-Spline Techniques, 1st ed. Springer-Verlag, 2002.
[31] S. Guarnieri, F. Piazza, A. Uncini, Multilayer feedforward networks with adaptive spline activation function, IEEE Transactions on Neural Network 10 (3) (1999) 672–683.
[32] E. Catmull, R. Rom, Ch. A class of local interpolating splines, in: Computer-Aided Geometric Design, Academic Press, New York, 1974, pp. 317–326.
[33] P. Borwein, T. Erdelyi, Polynomials and Polynomial Inequalities, Springer-Verlag, 1995.
[34] S. Kalluri, G.R. Arce, General class of nonlinear normalized adaptive filtering algorithms, IEEE Transactions on Signal Processing 47 (8) (1999) 2262–2272.
[35] V.J. Mathews, X. Zhenhua, A stochastic gradient adaptive filter with gradient adaptive step size, IEEE Transactions on Signal Processing 41 (6) (1993) 2075–2087.
[36] D.P. Mandic, J.A. Chambers, Recurrent Neural Networks for Prediction, John Wiley & Sons, 2001.
[37] D.P. Mandic, A.I. Hanna, M. Razaz, A normalized gradient descent algorithm for nonlinear adaptive filters using a gradient adaptive step size, IEEE Signal Processing Letters 8 (11) (2001) 295–297.
[38] D.P. Mandic, A generalized normalized gradient descent algorithm, IEEE Signal Processing Letters 11 (2) (2004) 115–118.
[39] L. Vecci, F. Piazza, A. Uncini, Learning and approximation capabilities of adaptive spline activation function neural networks, Neural Networks 11 (2) (1998) 259–270.

[40] G. Thimm, P. Moerland, E. Fiesler, The interchangeability of learning rate and gain in backpropagation neural networks, Neural Computation 8 (1996) 451–460.
[41] D.P. Mandic, J.A. Chambers, Relating the slope of the activation function and the learning rate within a recurrent neural network, Neural Computation 11 (1999) 1069–1077.
[42] F. Van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley, 1982.
[43] T.M. Panicker, V.J. Mathews, G.L. Sicuranza, Adaptive parallel-cascade truncated Volterra filters, IEEE Transactions on Signal Processing 46 (10) (1998) 2664–2673.
[44] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Transaction on Neural Networks 1 (1) (1990) 4–27.
[45] V. Su Lee Goh, D.P. Mandic, Recurrent neural networks with trainable amplitude of activation functions, Neural Networks 16 (2003) 1095–1100.
[46] A.D. Back, A.C. Tsoi, A simplified gradient algorithm for IIR synapse multilayer perceptrons, Neural Computation 5 (3) (1993) 456–462.