Hammerstein uniform cubic spline adaptive filters: Learning and convergence properties

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

Abstract

In this paper a novel class of nonlinear Hammerstein adaptive filters, consisting of a flexible memory-less function followed by a linear combiner, is presented. The nonlinear function involved in the adaptation process is based on a uniform cubic spline function that can be properly modified during learning. The spline control points are adaptively changed by using gradient-based techniques. This new kind of adaptive function is then applied to the input of a linear adaptive filter and it is used for the identification of Hammerstein-type nonlinear systems. In addition, we derive a simple form of the adaptation algorithm, an upper bound on the choice of the step-size and a lower bound on the excess mean square error in a theoretical manner. Some experimental results are also presented to demonstrate the effectiveness of the proposed method in the identification of high-order nonlinear systems.

1. Introduction

Many efforts have been done by the research community in the fields of modeling and identification of nonlinear systems. While a well established theory is available in literature for linear filters [1–3], in the nonlinear case only approximate models that are acceptable only in proximity of a specific operating point exist and a general theoretic framework is not available [1,4].

In order to model nonlinear systems, truncated Volterra series were introduced [5,6]. Volterra series, one of the most used black-box models, are a generalization of the Taylor series expansion based on convolutive 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 [7–10]. Other approaches, simpler than VAF, are based on an a priori fixed nonlinear expansion of the input data into a higher-dimensional space, where the identification problem can be solved in a linear manner. Examples of this kind of approaches are represented by Functional Link Adaptive Filters (FLAFs) [11,12] or Kernel Adaptive Filters (KAFs) [13].

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 [14]. The basic classes of block-oriented nonlinear systems are represented by the Wiener and Hammerstein models [4] and by those system architectures originated by the connection of these two classes according to different topologies (i.e. parallel, cascade, and feedback) [15]. More specifically, 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 [16–18]. The Hammerstein model, which consists of a cascade connection of a static nonlinear function followed by a LTI filter [19,20], 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 [1,21]. Of particular interest it has been the estimation of the specialized classes of Hammerstein systems. In fact, Hammerstein architectures can be successively used in several applications in all fields of engineering, from signal processing [22–24], to biomedical data analysis [25,26] or other applications, like hydraulics [27] or chemistry processes [28].

Actually, many existing methods for Hammerstein system identification are not adaptive [19,29]. On the other hand, adaptive methods, both parametric and non-parametric, are usually based on the use of some particular and fixed nonlinearities [22,30–32].

In this paper we present a NL block-oriented Hammerstein model, extending a new class of nonlinear adaptive filters called Spline Adaptive Filters (SAFs), belonging to the class of causal shift-invariant recursive nonlinear filters (like VAF, FLAF and KAF) and introduced very recently in [33] for the case of Wiener models. The proposed architecture, differently from recent existing approaches in which some particular and fixed nonlinearities must be chosen, is composed by an adaptable look-up table (LUT) addressed by the input and interpolated by a local low order polynomial spline curve with uniform knots followed by a linear combiner (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 to focus principally on some salient aspects related to SAF’s on-line adaptation, computational analysis and convergence properties. Some novel results in terms of an upper bound on the choice of the step-size are also derived. In addition a steady-state theoretical performance evaluation is deduced in terms of excess mean square error (EMSE) in a general manner, differently from previously works based on Gaussian input data [34], periodic sequences [12] or polynomial filters [23].

As a remark on the use of the spline function, it can be said that the adaptation of the control points is local in the sense that only a span (i.e. four control points) is adapted for each input sample and so the shape of the nonlinear function is slightly modified during learning. On the contrary, when a polynomial type of nonlinearity is used, even a change in only one polynomial coefficient implies changing of the whole nonlinearity shape. In addition, the regularization property common to most of the polynomial spline basis set, called variation diminishing property [35], ensures the absence of unwanted oscillations of the curve between two consecutive control points, as well as the exact representation of linear segments. These facts guarantee also the absence of chaotic behavior during the alternating learning phase. Moreover, even if the proposed approach does not benefit from cross-terms, like the VAF approach does, the cubic spline interpolation, due to its local interpolation scheme, is able to correctly identify high-order nonlinearities despite its low order polynomial and with a low computational cost.

This paper is organized as described in the following. Section 2 introduces a brief overview of the SAF. Section 3 describes the proposed novel Hammerstein Spline Adaptive Filter (HSAF) architecture and derives the learning rules for the adaptation of free parameters of the proposed Hammerstein model. Section 4 presents a theoretical proof of the convergence properties of the proposed architecture and shows a performance evaluation in terms of EMSE, while 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 weight vector, calculated according to some 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 \( \mathbf{s}_n \in \mathbb{R}^M = [s[n], s[n-1], \ldots, s[n-M+1]]^T \), or \( \mathbf{s}_{n-1} \in \mathbb{R}^M = [s[n-1], s[n-2], \ldots, s[n-M]]^T \). Note that in the absence of temporal index \( s \equiv s_n \) by default.

2. Brief review on SAF

For a complete introduction on spline adaptive filtering and spline interpolation, we refer to our recent paper [33].

In summary, with reference to Fig. 1, in order to compute the linear filter input \( s[n] = \varphi(x[n]) \), that is a uniform spline interpolation of some adaptive control points contained in a LUT, we have to determine the explicit dependence between the input signal \( x[n] \) and \( s[n] \). This can be easily done by considering that \( s[n] \) is a function of two local parameters \( u_i \) and \( i \) which depend on \( x[n] \). In the simple case of uniform spacing of knots and a third-order curve interpolation adopted in this work, the computation procedure for the determination of the span index \( i \) and the local parameters \( u_i \) can be expressed by the following equations [36]:

\[
\begin{align*}
\mathbf{u}_n &= \frac{x[n]}{\Delta x}, \\
i &= \left\lceil \frac{x[n]}{\Delta x} \right\rceil + \frac{Q - 1}{2},
\end{align*}
\]

where \( \Delta x \) is the uniform space between knots, \( \lceil \cdot \rceil \) is the floor operator and \( Q \) is the total number of control points. The second term in the second equation is an offset value needed to force \( i \) to be always nonnegative. Note that the index \( i \) really depending from time \( n \), i.e. \( i_n \); for simplicity of notation we adopt the convention \( i_n \equiv i \).

Referring to the top of Fig. 2 and to [33], the output of the nonlinearity can be evaluated as

\[
s[n] = \varphi_i(u_i) = u_i^T \mathbf{C}_q[n],
\]

where, considering a third-order spline basis, the matrix \( \mathbf{C} \in \mathbb{R}^{4 \times 4} \) is a pre-computed matrix, usually called spline basis matrix, the vector \( \mathbf{u}_i \) is defined as \( \mathbf{u}_i \in \mathbb{R}^{4 \times 1} = [u_i^1, u_i^2, u_i^3, u_i^4]^T \), and the vector \( \mathbf{q}_n \) contains the control points at instant \( n \) and is defined by \( \mathbf{q}_n \in \mathbb{R}^{4 \times 1} = [q_k, q_{k+1}, q_{k+2}, q_{k+3}]^T \), where \( q_k \) is the \( k \)-th entry in the LUT. We call the two blocks expressed by (1) and (2) as \( S_1 \) and \( S_2 \), respectively (see Fig. 2).

It could be very important to evaluate the derivative of (2) with respect to its input. It is easily evaluated in

\[
\varphi_i'(u_i) = u_i^T \mathbf{C}_q[n],
\]

where \( \mathbf{u}_i \in \mathbb{R}^{4 \times 1} = [3u_i^2, 2u_i, 1, 0]^T \).

3. Hammerstein spline adaptive filter

There exist several ways to approach the Hammerstein identification problem [37]. In this paper we use the standard stochastic gradient method. With reference to Figs. 1 and 2, let us pose \( \varepsilon[n] \) the output error that is defined as

\[
\varepsilon[n] = d[n] - y[n] = d[n] - \mathbf{w}_n^T \mathbf{s}_n,
\]

where \( \mathbf{s}_n \in \mathbb{R}^{M \times 1} = [s[n], s[n-1], \ldots, s[n-M+1]]^T \) and each sample \( s[n-k] \) is evaluated by using (2).

The on-line learning algorithm can be derived by considering the cost function \( J(\mathbf{w}_n, \mathbf{q}_n) = E[\varepsilon[n]^2] \). As usual, this CF is approximated by considering only the instantaneous error

\[
J(\mathbf{w}_n, \mathbf{q}_n) = \varepsilon[n]^2.
\]

For the minimization of (5), we proceed by applying the stochastic gradient adaptation. At the time-index \( n \) we can write

\[
\frac{\partial J(\mathbf{w}_n, \mathbf{q}_n)}{\partial \mathbf{w}_n} = \frac{-2\varepsilon[n] \partial y[n]}{\partial \mathbf{w}_n},
\]

where \( y[n] = \mathbf{w}_n^T \mathbf{s}_n \). Hence (6) becomes

\[
\frac{\partial J(\mathbf{w}_n, \mathbf{q}_n)}{\partial \mathbf{q}_n} = \frac{-2\varepsilon[n] \partial y[n]}{\partial \mathbf{q}_n}.
\]

For the derivative computation of (5) with respect to the control points \( \mathbf{q}_n \), we apply the chain rule

\[
\frac{\partial J(\mathbf{w}_n, \mathbf{q}_n)}{\partial \mathbf{q}_n} = 2\varepsilon[n] \frac{\partial \varphi_i(u_i)}{\partial \mathbf{q}_n} \frac{\partial \varphi_i(u_i)}{\partial \mathbf{w}_n} = -2\varepsilon[n] \frac{\partial y[n]}{\partial \mathbf{q}_n} \frac{\partial \varphi_i(u_i)}{\partial \mathbf{w}_n}.
\]

From (2), we have that \( \frac{\partial s_n}{\partial \mathbf{q}_n} = \mathbf{C} \mathbf{U}_{i,n} \), where \( \mathbf{U}_{i,n} \in \mathbb{R}^{4 \times M} = [u_{i,n}, u_{i,n-1}, \ldots, u_{i,n-M+1}]^T \) is a matrix which collects \( M \) past vectors \( u_{i,n-k} \), each one assuming the value of \( u_{i,n-k} \) if it was evaluated in the same span \( i \) of the current sample input \( x[n] \) or in an overlapped span, otherwise it is a zero vector. So we can write

\[
\frac{\partial J(\mathbf{w}_n, \mathbf{q}_n)}{\partial \mathbf{q}_n} = -2\varepsilon[n] \mathbf{C} \mathbf{U}_{i,n} \mathbf{w}_n.
\]

Note that, in (9) the update at time-index \( n \) is depending on past samples of the spline local parameter \( u_{i,n} \). The \( \mathbf{U}_{i,n} \) matrix, not so easy to be calculated, is essential to correctly evaluate the derivative \( \frac{\partial J}{\partial \mathbf{q}_n} \). This is similar to the back-propagation through time algorithm for neural networks [38]. This behavior is different from the Wiener case [33] where the derivative of the cost function with respect to the spline control points is depending on the current input sample alone.

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

\[
\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_n[n] \varepsilon[n] \mathbf{w}_n,
\]

\[
\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_q[n] \varepsilon[n] \mathbf{C} \mathbf{U}_{i,n} \mathbf{w}_n,
\]

where the parameters \( \mu_\varepsilon[n] \) and \( \mu_q[n] \), represent the learning rates for the weights and for the control points at time instant \( n \) respectively and, for simplicity, incorporate the others constant values. In the next section an analytical derivation of a bound for these 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

---

1 In this work we consider only real-valued variables.
involved curve span are updated. This is a consequence of the locality of the spline interpolation scheme.

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

**Algorithm 1.** Summary of the Hammerstein SAF-LMS algorithm.

```
Initialize: w_{-1} = \delta[n], q_{-1}
1: for n = 0, 1, \ldots \ do
2: \ u_n = x[n] / \Delta x - [x[n] / \Delta x]
3: \ i = [x[n] / \Delta x] + (Q - 1) / 2
4: \ \delta[n] = u_i^T q_{i,n}
5: \ y[n] = w_{i,n} T s_n
6: \ e[n] = d[n] - y[n]
7: \ w_{i,n+1} = w_{i,n} + \mu e[n] \delta[n] s_n
8: \ U_{i,n} = [u_{i,n}, u_{i,n-1}, \ldots, u_{i,n-M+1}]
9: \ q_{i,n+1} = q_{i,n} + \mu e[n] \delta[n] C^T U_{i,n} w_{i,n}
10: \ end for
```

The learning expressions (10) and (11) can be simply generalized to second order learning rules like quasi-Newton (QN) [39], Recursive Least Squares (RLS) [40], Affine Projection Algorithm (APA) [41] or other variants [2].

### 3.1. Computational analysis

For the computational cost analysis we refer to our previous work [33], where a comprehensive analysis is performed.

In summary, for each iteration only the i-th span of the curve is modified by calculating the quantities \( u_{i,n} \), \( i \) and the expressions \( u_i^T q_{i,n} \) and \( C^T U_{i,n} w_{i,n} \) appearing respectively in the 4-th and 8-th lines in Algorithm 1. However, most of the computations can be done through the re-use of past calculations. 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 [36]). 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

In order 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 \( \phi_i(u_n) \) [42]. For this purpose, it is useful to adopt an adaptive learning rate that minimizes the instantaneous output error of the filter [43].

Fig. 3 shows the model to be identified (denoted with \( \phi \)) and the adaptive architecture adopted. The convergence properties can be achieved by performing a Taylor expansion of the output 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 \( 5 \) 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. However, the adaptation can be performed in two separate phases. For example, the filter weights coefficients \( \mathbf{w} \) can
be preemptively adapted in the first phase of learning. In this way, the hypothesis that the change in weights can be minimal during the last phase of adaptation can be considered true.

4.1. Stability

The convergence property of the (10) can be derived from the Taylor series expansion of the error $e[n+1]$ around the instant $n$, stopped at the first order [44,45]

$$e[n+1] = e[n] + \frac{\partial e[n]}{\partial w_n} \Delta w_n + \text{h.o.t.}, \quad (12)$$

where h.o.t. means high order terms. Now, using (4) and (10), we derive

$$\frac{\partial e[n]}{\partial w_n} = -s_n^T, \quad (13)$$

$$\Delta w_n = \mu_w[n] e[n] s_n. \quad (14)$$

Substituting (13) and (14) in (12), we can obtain after simple manipulations

$$e[n+1] = (1 - \mu_w[n] \| s_n \|^2) e[n]. \quad (15)$$

In order to ensure the convergence of the algorithm, we desire that $|e[n+1]| < |e[n]|$. This aim is reached if the following relation holds

$$|1 - \mu_w[n] \| s_n \|^2| < 1, \quad (16)$$

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

$$0 < \mu_w[n] < \frac{2}{\| s_n \|^2}. \quad (17)$$

Note that in (17) all quantities are positive, thus no positivity constrains are needed. Now it is $s[n] = \varphi(x[n], q_u)\hat{y}[n]$, since expanding the function $\varphi(\cdot)$ in Taylor series, we obtain

$$s[n] = \varphi(x[n], q_u) = \varphi(0) + u_0 \varphi'(0). \quad (18)$$

Now using the expressions (2) and (3), it is possible to derive

$$\varphi'(0) = c_i q_{i,n}, \quad (20)$$

where $c_i \in \mathbb{R}^{1 \times 4}$ is the $k$-th row of the $C$ matrix.

Hence, let us pose $u_n = [u_n, u_{n-1}, \ldots, u_{n-M+1}]^T$ the vector that collects the variable $u_n$ at temporal instants $n, n-1, \ldots, n-M+1$ and $Q_{n} = 4 \times M$ matrix collecting the $q_{i,n}$ vectors at temporal instants $n, n-1, \ldots, n-M+1$, it is possible to write (18) in a vectorial form as

$$s_n \approx Q_{n}^T c_i^T + u_n \odot Q_{n}^T c_i^T, \quad (21)$$

where $\odot$ is the element-wise product. Thus Eq. (17) becomes

$$0 < \mu_w[n] \leq \frac{2}{\| Q_{n}^T c_i^T + u_n \odot Q_{n}^T c_i^T \|^2}. \quad (22)$$

Constraint (22) shows that the convergence properties depend on the particular basis of spline function, due to the terms $c_i$ and $c_i$ and the spacing between knots, since the terms $u_n$ are depending from $\Delta x$.

In order to stabilize the expression in (17) 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

$$\mu_w[n] = \frac{2}{\| s_n \|^2 + \delta_w[n]}. \quad (23)$$

The regularizing term $\delta_w[n]$ can be adapted by using a gradient descent approach [45,46]:

$$\delta_w[n+1] = \delta_w[n] - \eta_w \nabla_{\delta_w}[w_n, q_{i,n}], \quad (24)$$

where $\eta_w$ is the learning rate. For the evaluation of the gradient in (24), we use the chain rule

$$\nabla_{\delta_w}[w_n, q_{i,n}] = \frac{\partial f[w_n, q_{i,n}]}{\partial \delta_w[n]} = \frac{\partial f[w_n, q_{i,n}]}{\partial e[n]} \cdot \frac{\partial e[n]}{\partial w_n} \cdot \frac{\partial w_n}{\partial \delta_w[n]} \quad (25)$$

Hence, we derive

$$\delta_w[n+1] = \delta_w[n] - \eta_w \frac{e[n] e[n-1] s_n^T s_{n-1}}{\| s_n \|^2 + \delta_w[n-1]^2}. \quad (26)$$
Note that in (26) the terms \(s_n\) and \(s_{n-1}\) depend on the value of the nonlinear adaptive function and its derivative in the origin \(\varphi(0)\) and \(\varphi'(0)\), using (18) or (21).

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

\[
e[n+1] = e[n] + \frac{\partial e[n]}{\partial q[n]} \Delta q[n] + \text{h.o.t.}
\]

and, from (4) and (11), the equations

\[
\frac{\partial e[n]}{\partial q[n]} = -C^T U_n w_n, 
\]

\[
\Delta q[n] = \mu_q[n] e[n] C^T U_n w_n. 
\]

Hence we derive, after simple manipulations

\[
e[n+1] = [1 - \mu_q[n]] \|C^T U_n w_n\|^2 e[n].
\]

In order to ensure the convergence of the algorithm, imposing the uniform convergence of (30) as done in (16), we obtain

\[
|1 - \mu_q[n]| \|C^T U_n w_n\|^2 < 1,
\]

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

\[
0 < \mu_q[n] < \frac{2}{\|C^T U_n w_n\|^2}. 
\]

Eqs. (17) and (32) impose on the learning rates simple constraints. A more restrictive constraint must be satisfied by \(\mu_w[n]\) and \(\mu_q[n]\) simultaneously. In order to obtain such a condition, rewrite the output error expansion as

\[
e[n+1] = e[n] + \frac{\partial e[n]}{\partial \mathbf{w}_n} \mathbf{w}_n = \text{const} \Delta \mathbf{w}_n + \frac{\partial e[n]}{\partial \mathbf{q}_n} \mathbf{q}_n = \text{const} \Delta \mathbf{q}_n. 
\]

Using (13), (14), (28) and (29), Eq. (33) can be rewritten as

\[
e[n+1] = e[n][1 - \mu_w[n]|s_n|^2 - \mu_q[n]|C^T U_n w_n|^2].
\]

Imposing once again the condition \(|e[n+1]| < |e[n]|\), we finally obtain the constraint

\[
0 < \mu_w[n]|s_n|^2 + \mu_q[n]|C^T U_n w_n|^2 < 2. 
\]

4.2. Mean square performance

The aim of this paragraph is to study the mean square performance of the proposed architecture at steady-state. In particular we are interested in the derivation of the excess mean square error (EMSE) of the nonlinear adaptive filter [2]. Even in this case the analysis is performed in a separate manner, considering first the case of the linear filter and then the nonlinear spline function. Note that for EMSE evaluation we are interested in steady-state, so the fixed sub-system can be considered adapted. In the following we denote with \(\varepsilon[n]\) the a priori error of the whole system in Fig. 3, with \(\varepsilon_w[n]\) the a priori error when only the linear filter is adapted while the spline control points are fixed and similarly with \(\varepsilon_q[n]\) the a priori error when only the control points are adapted while the linear filter is fixed. In order to make tractable the mathematical derivation, some assumption must be introduced:

A1. The noise sequence \(v[n]\) is i.i.d. with variance \(\sigma_v^2\) and zero mean.

A2. The noise sequence \(v[n]\) is independent of \(x_n, s_n, e[n], \varepsilon_w[n]\) and \(\varepsilon_q[n]\).

Let us define the weight error vector as \(\psi_w[n]\ = w_0 - w_n\). From (10) we can write

\[
\psi_w[n] = \psi_w[n-1] - \mu_w[n][e[n] s_n]. 
\]

Now let us evaluate the a priori error \(\varepsilon_w[n] = d[n] - y[n]\) when only the linear filter with parameters \(w_n\) is adapted. We obtain \(\varepsilon_w[n] = w_0^T r_n - w_n^T s_n \approx (w_0 - w_n)^T s_n = \psi_w[n]^T s_n\). (37)

where \(r_n = [r[n], r[n-1], \ldots, r[n-M+1]]^T\) and \(r_n = \varphi(x[n])\) is the output of the reference nonlinearity. In (37) we use the approximation \(r_n \approx s_n\), and hence \(q[n] \approx q_0\), that can be considered a reasonable assumption at steady-state.

Now evaluating the energies of both sides of (36), we obtain

\[
\|\psi_{w[n+1]}\|^2 = \|\psi_{w[n]}\|^2 - 2\mu_w[n]\|e[n]\|\varepsilon_w[n]\| + \mu_w[n]\|e[n]\|^2 |s_n|^2. 
\]

Taking the expectation of both sides of (38) and noting that, at steady-state for \(n \rightarrow \infty\), is \(E[\|\psi_{w[n]}\|^2] = \|\psi_{w[n]}\|^2\), it can be obtained

\[
2E[\|e[n]\|\varepsilon_w[n]] = \mu_w[n]E[\|e[n]\|^2 |s_n|^2]. 
\]

Remembering that \(e[n] = \varepsilon_w[n] + v[n]\), where \(v[n]\) is an additive noise on reference signal (see Fig. 3) that is uncorrelated with the a priori error for assumption A2 [47], we can easily derive

\[
E[\|e[n]\|\varepsilon_w[n]] = E[\varepsilon_w[n] + v[n]]\varepsilon_w[n] = E[\varepsilon_w[n]]. 
\]

Since \(|s_n|^2\) is the steady-state \(L_2\) norm, and justified from the fact that at steady-state the error is very small, we have

\[
E[\|e[n]\|^2 |s_n|^2] = E[(\varepsilon_w[n] + v[n])^2 |s_n|^2] = |s_n|^2 E[\varepsilon_w[n] + v[n]]^2. 
\]

\[
= |s_n|^2 \sigma_v^2 + \sigma_y^2. 
\]

where \(\sigma_y^2\) is the variance of the noise \(v[n]\) (assumption A1).

Inserting (40) and (41) in (39), it is possible to derive

\[
2E[\varepsilon_w[n]^2] = \mu_w[n]E[\varepsilon_w[n]^2] + \sigma_y^2 |s_n|^2. 
\]

Denoting with \(\zeta_w = \lim_{n \rightarrow \infty} E[\varepsilon_w[n]^2]\) the EMSE on the linear filter, from (42) we finally obtain

\[
\zeta_w = \frac{\mu_x[n] \sigma_y^2 |s_n|^2}{2 - \mu_w[n] |s_n|^2}. 
\]

For the consistency of \(\zeta_w\) in (43), it is necessary that the EMSE is a positive value. For this purpose, we impose the positivity of the denominator of (43). This easily leads to Eq. (17).

If the learning rate \(\mu_x[n]\) is very small, the second term in the denominator of (43) can be neglected, so the
expression of EMSE reduces to
\[ \zeta_{w} = \mu_{q}[n] \frac{\|n\|}{2} E[\|q\|^2]. \]  
(44)

It is interesting to note that the derivatives of Eqs. (43) and (44) with respect to the learning rate \( \mu_{q}[n] \) are always positive, hence the EMSE has not a minimum value and an optimum value for \( \mu_{q}[n] \) is not existing.

With a similar reasoning, starting from the definition of control-point error vector \( v[n] = q_{0} - q_{n} \) and (11), we obtain
\[ v[n+1] = v[n] - \mu_{q}[n] e[n] C_{i} U_{i,n} w_{n}. \]  
(45)

Evaluating now the \textit{a priori} error \( e_{q}[n] \), we obtain
\[ e_{q}[n] = w_{n}^{T} r[n] - w_{n}^{T} s_{n} \approx w_{n}^{T} (r_{n} - s_{n}) = w_{n}^{T} U_{i,n} w_{n}, \]  
(46)

in which we use the approximation \( w_{0} \approx w_{n} \), that can be considered a reasonable assumption at steady-state.

Evaluating now the energies of both sides of (45) we obtain
\[ \|v[n+1]\|^2 = \|v[n]\|^2 - 2 \mu_{q}[n] e[n] e[n] w_{n} + \mu_{q}[n] e[n] e[n] C_{i} U_{i,n} w_{n}. \]  
(47)

Taking the expectation and remembering that, for \( n \to \infty \), is \( E[\|v[n+1]\|^2] = E[\|v[n]\|^2] \), it can be obtained
\[ 2 E(e[n] e[n]) = \mu_{q}[n] E[\|C_{i} U_{i,n} w_{n}\|^2]. \]  
(48)

For assumption A2, it is possible to derive
\[ E(e[n] e[n]) = E(e[n] e[n] + v[n] e[n]) = E(e[n] e[n]). \]  
(49)

Since \( \|C_{i} U_{i,n} w_{n}\|^2 \) is the steady-state L2 norm, and justified from the fact that at steady-state the error is very small, we have
\[ E(e[n] e[n]) = E(e[n] e[n] + v[n] e[n]) \approx \|C_{i} U_{i,n} w_{n}\|^2 [E(e[n] e[n]) + \sigma_{v}^2]. \]  
(50)

where \( \sigma_{v}^2 \) is the variance of the noise \( v[n] \) (assumption A1). Hence, we can derive
\[ 2 E(e[n] e[n]) = \mu_{q}[n] E(e[n] e[n]) + \sigma_{v}^2 \|C_{i} U_{i,n} w_{n}\|^2. \]  
(51)

Denoting with \( \zeta_{q} = \lim_{n \to \infty} E(e[n] e[n]) \) the EMSE concerning the spline control points \( q \), from (42) we finally obtain
\[ \zeta_{q} = \mu_{q}[n] \frac{\|C_{i} U_{i,n} w_{n}\|^2}{2 - \mu_{q}[n] \|C_{i} U_{i,n} w_{n}\|^2}. \]  
(52)

For the consistency of \( \zeta_{q} \) in (52), we impose the positivity of the denominator of (52). This easily leads to (32). If the learning rate \( \mu_{q}[n] \) is very small, (52) reduces to
\[ \zeta_{q} = \frac{\mu_{q}[n]}{2} \frac{\|C_{i} U_{i,n} w_{n}\|^2}{\sigma_{v}^2 \|C_{i} U_{i,n} w_{n}\|^2}. \]  
(53)

A similar conclusion on the nonexistence of an optimum value for \( \mu_{q}[n] \) can be done.

Now it can be considered that the \textit{a priori} error \( e[n] \) of the whole system is due to the error \( e_{w}[n] \) due to the filter taps and the error \( e_{q}[n] \) due to the control points. In fact, using (37) and (46), we obtain
\[ e[n] = d[n] - y[n] = w_{0}^{T} r[n] - w_{n}^{T} s_{n} \]
\[ = (w_{0}^{T} r[n] - w_{n}^{T} s_{n}) + (w_{0}^{T} r[n] - w_{n}^{T} s_{n}) \]
\[ = (w_{0}^{T} r[n] - w_{n}^{T} s_{n}) + (w_{0}^{T} r[n] - w_{n}^{T} s_{n} + (w_{0}^{T} r[n] - w_{n}^{T} s_{n}) \]
\[ = w_{0}^{T} U_{i,n} C_{i} (q_{0} - q_{n}) + (w_{0}^{T} r[n] - w_{n}^{T} s_{n}) \]
\[ \approx w_{0}^{T} U_{i,n} C_{i} (q_{0} - q_{n}) + (w_{0}^{T} r[n] - w_{n}^{T} s_{n}) \]
\[ = w_{0}^{T} U_{i,n} C_{i} (q_{0} - q_{n}) + e_{w}[n] + e_{q}[n]. \]  
(54)

Hence
\[ \zeta = \lim_{n \to \infty} E[e_{w}[n] + e_{w}[n]^2] \]
\[ = \lim_{n \to \infty} E[e_{w}[n] + e_{w}[n]^2] + \lim_{n \to \infty} E[e_{w}[n] e_{w}[n]] \]
\[ = \zeta_{w} + \zeta_{q} + 2 \zeta_{w} \zeta_{q} \]
(55)

where \( \zeta_{w} = \lim_{n \to \infty} E[e_{w}[n] e_{w}[n]] \) denotes the cross-EMSE. When errors \( e_{w}[n] \) and \( e_{q}[n] \) could be considered statistically independent and zero mean then \( \zeta_{w} = 0 \), and the EMSE is the lower possible. Eq. (55) justifies the following

Property 1. The EMSE \( \zeta \) of a Hammerstein spline adaptive filter verifies the condition
\[ \zeta \geq \zeta_{w} + \zeta_{q}. \]  
(56)

From experimental tests we have seen that the simulated EMSE is very close to the summation on the right side of (56) (see Experiment 2), so the errors \( e_{w}[n] \) and \( e_{q}[n] \) are close to be uncorrelated.

5. Experimental results

In order to validate the proposed nonlinear adaptive filtering solution, several experiments were performed. Experimental tests address high-order nonlinear system identification problem. Uniform cubic spline nonlinearities have been used, as they represent a good compromise between flexibility and approximation behavior. Comparisons with two different approaches, in which the non-linearity is implemented by a polynomial function, were also performed. The first of these approaches is the Hammerstein architecture proposed by Stenger and Kellerman in [48], where the linear filter coefficients and the polynomial coefficients are alternatively adapted by using an LMS algorithm. The second one is the particular approach proposed by Jeraj and Mathews in [30] in which a partial orthogonalization of the inputs signal is performed and the free parameters are adapted all together. Since the latter approach considers an IIR linear filter, in order to perform a substantial comparison between all architectures, only the MA part of the linear filter is considered.

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 six experiments. Due to the local scheme of spline interpolation, the possibility of chaotic behavior, as underlined in Section 1, could be excluded: in fact we never observed such a chaotic behavior in experimental sessions, and always a good result is obtained if the learning rates are adequately chosen.

As explained in Section 4, although the cost function may not admit a unique minimum value and thus the

\[ \text{Some Matlab source codes, implementing the experimental results, are available at the following web page: http://ispac.ing.uniroma1.it/ HSAF.html.} \]
convergence toward the optimum cannot be guaranteed, no particular attention should be posed on the choice of the initial conditions for the filter weights and spline control points. Without any specific a priori knowledge, a reasonable choice of initial conditions that have always guaranteed excellent results is $w_{-1} = a0$ for filter weights with $0 < a < 1$ and $\mathbf{q}_{-1} = [0,\ldots,0]^T$, while spline control points $\mathbf{q}_{-1}$ are initialized as a straight line with a unitary slope, i.e. we begin to adapt a linear filter. We have to emphasize that such a choice, with a positive slope of the nonlinearity, avoids the convergence toward $-w_0$ and $-q_0$, preventing a sign ambiguity.

5.1. Experiment 1

A first experiment is performed in order to show the convergence behavior of the SAF illustrated in Section 4. The experiment consists in the identification of an unknown Hammerstein system composed by a linear component $w_0 = [0.6, -0.4, 0.25, -0.15, 0.1, -0.05, 0.001]^T$ and a nonlinear memoryless target function implemented by a 23-points length LUT $\mathbf{q}_0$ and interpolated by a uniform third degree spline with an interval sampling $\Delta x = 0.2$ defined as

$$\mathbf{q}_0 = [-2.2, -2.0, -1.8, \ldots, -1.0, -0.8, -0.91, -0.40, -0.20, 0.05, 0.0, -0.40, 0.58, 1.0, 1.0, 1.2, 1.4, \ldots, 2.2]$$

The input signal $x[n]$ consists of 30,000 samples of the signal generated by the following relationship:

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

(57)

where $\xi[n]$ is a zero mean white Gaussian noise with unitary variance and $0 \leq a < 1$ is a parameter that determines the level of correlation between adjacent samples. Experiments were conducted with $a$ set to 0.1 and 0.95. In addition it is considered as an additive white noise $\nu[n]$ such that the signal to noise ratio is SNR=60 dB, where $\text{SNR} = 10 \log_{10}[E[d^2[n]]/E[\nu^2][n]]$ (see Fig. 3 for symbols definition). The learning rates are set to $\mu_w = \mu_q = 0.1$. The filter weights are initialized using $\alpha = 0.1$.

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 the most important spline control points. Different choices for the $C$ matrix are possible: the most suitable for this kind of applications are the so-called B-spline and Catmull–Rom spline (or simply CR-spline), introduced in [33] and reported below

$$C_B = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_C = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 2 & 3 & 4 \end{bmatrix}$$

Both B-spline and CR-spline basis are used in the experiment, obtaining similar results.

In addition Fig. 4 shows a comparison of the MSE for the proposed experimental test with the two different choices of the parameter $a$. It is simple to note that, at steady state, the performance of the HSADF algorithm reaches the value of the noise power. Fig. 5 shows a comparison of the profile of the nonlinearity used in the model (the red dashed line) and the profile of the adapted nonlinearity obtained by spline interpolation in HSADF (the solid black line) when CR-spline is used and $a=0.95$. Note that the adapted nonlinearity is well overlapping the model one, as it is also evident from Table 2.

Table 2

<table>
<thead>
<tr>
<th>$i$</th>
<th>$q_i$</th>
<th>CR-spline</th>
<th>B-spline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Variance $\times 10^{-5}$</td>
<td>Mean</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>0.996</td>
<td>11.961</td>
</tr>
<tr>
<td>8</td>
<td>-0.80</td>
<td>0.796</td>
<td>8.413</td>
</tr>
<tr>
<td>9</td>
<td>-0.91</td>
<td>0.906</td>
<td>10.280</td>
</tr>
<tr>
<td>10</td>
<td>-0.40</td>
<td>0.398</td>
<td>1.683</td>
</tr>
<tr>
<td>11</td>
<td>-0.20</td>
<td>0.199</td>
<td>0.476</td>
</tr>
<tr>
<td>12</td>
<td>0.05</td>
<td>0.049</td>
<td>9.059 $\times 10^{-8}$</td>
</tr>
<tr>
<td>13</td>
<td>0.00</td>
<td>0.000</td>
<td>6.177 $\times 10^{-9}$</td>
</tr>
<tr>
<td>14</td>
<td>-0.40</td>
<td>0.398</td>
<td>2.294</td>
</tr>
<tr>
<td>15</td>
<td>0.58</td>
<td>0.577</td>
<td>3.834</td>
</tr>
<tr>
<td>16</td>
<td>1.00</td>
<td>0.905</td>
<td>12.571</td>
</tr>
</tbody>
</table>

Fig. 4. Comparison of MSE for experiment 1 using model (57) with $a=0.1$ and 0.95 respectively.
5.2. Experiment 2

In this experiment we prove the validity of (56) in Property 1. The test consists in the identification of the same system of the previous experimental test. The input signal is (57) with $a=0.1$. For simplicity we use $\mu_0[n] = \mu_q[n] = \mu$, chosen as the following 12 values:

$$\mu = \{0.001, 0.01, 0.02, ..., 0.09, 0.1, 0.15\}.$$

Fig. 6 shows the curves of the simulated EMSE and the theoretical one from Eq. (56) in the case of B-spline with $\Delta x = 0.2$. The simulated EMSE shown in Fig. 6 is obtained as a mean over the last 100 samples of the mean square error $e^2[n]$ averaged over a hundred simulations. It is clear that the simulated EMSE is lower-bounded from the theoretical one, as predicted and the distance between the two curves is very small. Similar results can be provided using a CR-spline.

5.3. Experiment 3

A third experimental set-up, drawn from [49], consists in the identification of a nonlinear dynamic system composed by two blocks. The first block is the following nonlinearity:

$$s[n] = \sqrt{x[n]}.$$

while the second block is the FIR filter with taps

$$h = [1, 0.75, 0.5, 0.25, 0, -0.25]^T.$$

The learning rates are set to $\mu_0 = \mu_q = 0.1$ and B-spline basis with $\Delta x = 0.2$ is used. We compare the proposed HSAF architecture with the polynomial Hammerstein architecture in [48] with the learning rate set to $\mu_p = 10^{-3}$ and the approach proposed in [30] by Jeraj and Mathews with $\mu = 10^{-5}$ implementing both a third order polynomial nonlinearity. The filter weights are initialized using $\alpha = 0.1$. For all architecture we use $M = 15$ parameters and a total of $5 \times 10^4$ samples are used. A comparison of the averaged MSE over 100 trials for the three architectures is reported in Fig. 7, where is clearly observable the goodness of the proposed approach.

5.4. Experiment 4

A fourth experimental set-up, drawn from [49], consists in the identification of a nonlinear dynamic system composed by two blocks. The first block is the following nonlinearity:

$$s[n] = \frac{1}{8} + \lfloor 2x[n] - 1 \rfloor,$$

where $\lfloor \cdot \rfloor$ is the floor operator and the second block is the FIR filter in (59) scaled of a factor 0.5. The learning rates are set to $\mu_0 = 0.01$ and $\mu_q = 0.02$ and B-spline basis with $\Delta x = 0.2$ is used. We compare the proposed HSAF architecture with the polynomial Hammerstein architecture in [48] with the learning rate set to $\mu_p = 10^{-3}$ and the approach proposed in [30] with $\mu = 10^{-5}$ implementing both a third order polynomial nonlinearity. The filter
weights are initialized using $\alpha = 0.1$. For all architecture we use $M=15$ parameters and a total of $5 \times 10^4$ samples are used. A comparison of the averaged MSE over 100 trials for the two architectures is reported in Fig. 8.

5.5. Experiment 5

A fifth experimental set-up is drawn from [20] and consists in the identification of the following nonlinearity:

$$s[n] = x[n] - 0.3x^2[n] + 0.2x^3[n]$$  \hspace{1cm} (61)

followed by the 8-th order IIR filter

$$H(z) = \frac{1 - 1.8000z^{-1} - 1.6200z^{-2} - 1.4500z^{-3} + 0.6561z^{-4}}{1 - 0.2314z^{-1} + 0.4318z^{-2} - 0.3404z^{-3} + 0.5184z^{-4}}$$  \hspace{1cm} (62)

The learning rates are set to $\mu_w = 0.01$ and $\mu_q = 0.02$ and B-spline basis with $\Delta x = 0.2$ is used. We compare the proposed HSAF architecture with the polynomial Hammerstein architecture in [48] with the learning rate set to $\mu_w = 0.01$ and the approach proposed in [30] with $\mu = 10^{-4}$ implementing both a third order polynomial nonlinearity. The filter weights are initialized using $\alpha = 0.1$. For all architecture we use $M=15$ parameters and a total of $5 \times 10^4$ samples are used. A comparison of the averaged MSE over 100 trials is reported in Fig. 9, where it can be seen that all three approaches perform in the same manner. This particular behavior is due to the simple nonlinearity chosen as a third order polynomial, since the number of free parameter is the same in all methods and it is very small.

5.6. Experiment 6

In order to verify that the robustness of the proposed approach with respect to the increasing of the nonlinear order, a last experiment is performed. The experiment consists of a comparison of the HSAF structure, with respect to the pure polynomial method [48] and the one proposed in [30], in the case of identification of a polynomial nonlinearity of various order $P$ in the form $\phi(x) = \sum_{k=1}^{P} b_k x^k$ and a FIR filter with taps $\mathbf{w}_0 = [1, -0.4, 0.25, -0.15, 0.1, -0.05, 0.001]^T$ (the reader can also do other random choice). The input signal is a white Gaussian noise with zero mean and unitary variance; a total of $5 \times 10^4$ samples are used. In addition it is considered an additive white noise $\mathbf{v}[n]$ such as the SNR is 50 dB. Six different polynomials (from 3-rd to 15-th order) are used. In the adaptive algorithms we used a number of linear coefficients $M = 15$, and the order of the polynomial nonlinearity is set exactly equal to the one of the model, while a B-spline basis is used. The learning rates are set to $\mu_w = 0.01$ and $\mu_q = 0.02$ for HSAF, $\mu_w = \mu_q = 0.01$ for the polynomial architecture and $\mu = 10^{-4}$ for the Jeraj and Mathews algorithm. The filter weights are initialized using $\alpha = 0.1$ and 100 trials of the algorithms are performed. Comparisons of steady-state MSE, evaluated as a mean of the last 100 mean square errors, are shown in Table 3.

This table demonstrates that the proposed approach can take some advantages in the case of high-order nonlinearities, since the performances in terms of MSE are quite stable with respect to an increase in the nonlinear order, while the performances of the other approaches decrease consequently. This result can be explained by considering the spline’s self-regularization capabilities related to (1) the local adaptation and (2) the

![Fig. 8. MSE of the proposed approach compared different approaches in experiment 4.](image1)

![Fig. 9. Comparison MSE for the different Hammerstein SAF in experiment 5.](image2)

<table>
<thead>
<tr>
<th>Order</th>
<th>MSE [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HSAF</td>
</tr>
<tr>
<td>3</td>
<td>-49.685</td>
</tr>
<tr>
<td>5</td>
<td>-49.192</td>
</tr>
<tr>
<td>7</td>
<td>-49.618</td>
</tr>
<tr>
<td>10</td>
<td>-49.791</td>
</tr>
<tr>
<td>12</td>
<td>-49.426</td>
</tr>
<tr>
<td>15</td>
<td>-49.348</td>
</tr>
</tbody>
</table>
principle of minimal disturbance since a new input data does not modify the whole nonlinear shape but only a well-defined portion [50,51]. On the contrary in using a polynomial type of nonlinearity, the adaptation concerns all coefficients that have imply changing of the whole nonlinearity shape.

6. Conclusion

This paper introduces a novel nonlinear Hammerstein adaptive filtering model, where the nonlinearity is implemented using spline functions. Splines are flexible nonlinear functions, whose shape can be modified during the simple learning process using gradient-based techniques. A uniform knots spacing is used.

The adaptation of the proposed nonlinear spline adaptive filter is based on the LMS algorithm. In addition a constraint on the choice of the learning rate, in order to assure the algorithm convergence, and a lower bound on the architectural EMSE are also theoretically derived.

Several experimental tests have demonstrated the effectiveness of the proposed approach. In particular HSAF can reach good performance with low computational cost also in the case of high-order nonlinearities, against other existing approaches.

Acknowledgments

The authors would like to thank all anonymous reviewers for their precious and helpful comments and suggestions.

References


