



Fast communication

Nonlinear system identification using IIR Spline Adaptive Filters



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 10 February 2014

Received in revised form

22 August 2014

Accepted 31 August 2014

Available online 6 September 2014

Keywords:

Nonlinear adaptive filter

Spline adaptive filter

Wiener system identification

Spline interpolation

Least mean square

ABSTRACT

The aim of this paper is to extend our previous work on a novel and recent class of nonlinear filters called Spline Adaptive Filters (SAFs), implementing the linear part of the Wiener architecture with an IIR filter instead of an FIR one. The new learning algorithm is derived by an LMS approach and a bound on the choice of the learning rate is also proposed. Some experimental results show the effectiveness of the proposed idea.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

In the last few decades several researchers have made many efforts towards nonlinear adaptive filtering theory and applications [1,2]. Unfortunately, differently from the linear case [3], a general theoretic framework is not yet available for the nonlinear one.

In order to model nonlinear systems, Volterra series [4] were introduced so far. Due to the large number of free parameters required, the Volterra Adaptive Filter (VAF) is generally used only in situations of mild nonlinearity [4,5]. Approaches based on simplified VAF implementing fast affine projection algorithm (FAPA) are often preferred [6,7]. Also neural networks (NNs) [8] represent a flexible tool to realize nonlinear filters, but this approach generally requires a high computational cost and shows some difficulties in adaptivity.

In practice, the so-called block-oriented representation, is used in nonlinear filtering, which consists of the connection of a linear time invariant (LTI) filter and a memoryless nonlinear functions. The basic classes of block-oriented nonlinear systems are represented by the Wiener model (a cascade of a linear LTI filter followed by a static nonlinear function) and the Hammerstein model (a cascade connection of a static nonlinear function followed by an LTI filter) and by those system architectures originated by the connection of these two classes according to different topologies (i.e. parallel, cascade, feedback etc) [2].

Very recently, we have proposed a novel block-oriented Wiener architecture, called Spline Adaptive Filters (SAFs) [9]. The proposed architecture is composed by a FIR filter followed by an adaptable look-up table (LUT) addressed by the linear combiner output and interpolated by a local low order polynomial spline curve. In [9] it is shown that SAF is able to well identify Wiener systems with high order nonlinearities.

In this brief paper we want to extend the SAF architecture proposed in [9], implementing the linear filter with an adaptive IIR architecture. The main advantage of using IIR filters over FIR filters is their efficiency in

* Corresponding author. Tel.: +39 06 44585869; fax: +39 06 4873300.

E-mail addresses: michele.scarpiniti@uniroma1.it (M. Scarpiniti),
daniilo.comminiello@uniroma1.it (D. Comminiello),
raffaele.parisi@uniroma1.it (R. Parisi),
aurelio.uncini@uniroma1.it (A. Uncini).

implementation: the matching of a particular specification can be accomplished with a lower number of parameters. The whole system will be adapted by a stochastic descent gradient using an output error-like method and a bound on the learning rate will be provided as well.

Since the LUT interpolated by a spline is a bounded function, the stability (in BIBO sense) of the proposed structure is guaranteed if the IIR part is stable. So in this work the problem of stability criteria, related to the poles of the IIR transfer function, is not addressed. Many authors have proposed in the literature different solutions to this problem [10,11], that can be simply imported in our architecture. However using the derived bound in conjunction with a simple heuristic parameters initialization, no particular stability issues have arisen during experimental results. In any case, the position of poles can be checked during the learning phase, and any unstable poles can easily be projected back inside the stable region to some appropriate location [10], with the drawback of a higher computational cost.

2. Spline interpolation

Splines are smooth parametric curves defined by interpolation of properly defined control points collected in a lookup table. Let $y[n] = \varphi(s[n])$ be some function to be estimated. The spline estimation provides an approximation $\varphi_{i_n}(u_n)$ based on two parameters u_n and i_n directly depending on $s[n]$. In the general case, given Q equispaced control points, the spline curve results as a polynomial interpolation through $Q - 1$ adjacent spans. In this specific application, we use cubic spline curves, so for each input occurrence $s[n]$ the spline is using four control points selected inside the lookup table. Two points are the adjacent control points on the left side of $s[n]$, while the other two points are the control points on the right side. The computation procedure for the determination of the span index i_n and the local parameters u_n can be expressed as follows [9]:

$$\begin{aligned} u_n &= \frac{s[n]}{\Delta x} - \left\lfloor \frac{s[n]}{\Delta x} \right\rfloor, \\ i_n &= \left\lfloor \frac{s[n]}{\Delta x} \right\rfloor + \frac{Q-1}{2}, \end{aligned} \quad (1)$$

where Δx is the uniform space between knots, $\lfloor \bullet \rfloor$ is the floor operator and Q is the total number of control points. For simplicity of notation in the following we will use $i_n \equiv i$.

The interpolated nonlinear output $y[n]$ can be easily evaluated by the following matrix product:

$$y[n] = \varphi_i(u_n) = \mathbf{u}_n^T \mathbf{C} \mathbf{q}_{i,n}, \quad (2)$$

where $\mathbf{u}_n \in \mathbb{R}^{4 \times 1} = [u_n^3, u_n^2, u_n, 1]^T$, $\mathbf{q}_{i,n} \in \mathbb{R}^{4 \times 1} = [q_i, q_{i+1}, q_{i+2}, q_{i+3}]^T$ with q_k be the k th control points and \mathbf{C} is a 4×4 matrix, depending on which spline basis is adopted to perform interpolation, usually B-spline or Catmul-Rom (CR) spline [9]. The derivative of (2) with respect the local abscissa u_n can be evaluated simply by

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

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

We refer to our previous work [9] for a more complete reference to this topic.

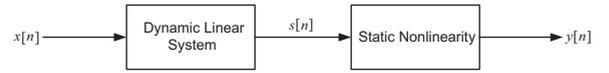


Fig. 1. Block diagram of a Wiener system.

3. IIR Wiener spline adaptive filter

With reference to Fig. 1, let the input–output relationship of the adaptive filter be

$$y[n] = \varphi(s[n], \mathbf{q}_{i,n}) = \mathbf{u}_n^T \mathbf{C} \mathbf{q}_{i,n}, \quad (4)$$

$$s[n] = \sum_{k=0}^{M-1} b_k[n] x[n-k] + \sum_{k=1}^N a_k[n] s[n-k], \quad (5)$$

where $b_k[n]$ and $a_k[n]$ are, respectively, the k th parameter of the MA and AR part of the IIR system at the time index n . M and N are the numbers of coefficients of the MA and AR parts, respectively. For simplicity, the filter taps vector \mathbf{w}_n is written as

$$\mathbf{w}_n \in \mathbb{R}^{(M+N) \times 1} = [b_0[n], b_1[n], \dots, b_{M-1}[n], a_1[n], \dots, a_N[n]]^T. \quad (6)$$

Denoting then with $\hat{\mathbf{x}}_n \in \mathbb{R}^{(M+N) \times 1} = [x[n], x[n-1], \dots, x[n-M+1], s[n-1], \dots, s[n-N]]^T$, the IIR filter output can be expressed as

$$s[n] = \mathbf{w}_n^T \hat{\mathbf{x}}_n. \quad (7)$$

In order to derive the LMS learning algorithm, let us consider the cost function

$$J(\mathbf{w}_n, \mathbf{q}_{i,n}) = e^2[n], \quad (8)$$

where $e[n] = d[n] - y[n]$ is the error signal and $d[n]$ is the reference signal. Hence the learning rule for the adaptation of the linear filter coefficients is given by

$$\begin{aligned} \Delta \mathbf{w}_n &= \nabla_{\mathbf{w}_n} J(\mathbf{w}_n, \mathbf{q}_{i,n}) = -2e[n] \nabla_{\mathbf{w}_n} y[n] \\ &= -2e[n] \left[\frac{\partial y[n]}{\partial u_n} \frac{\partial u_n}{\partial s[n]} \nabla_{\mathbf{w}_n} s[n] + \frac{1}{\Delta x} \nabla_{\mathbf{w}_n} s[n] \right], \end{aligned} \quad (9)$$

where the gradient vector $\nabla_{\mathbf{w}_n} s[n]$ is defined as

$$\nabla_{\mathbf{w}_n} s[n] = \left[\frac{\partial s[n]}{\partial b_0[n]}, \dots, \frac{\partial s[n]}{\partial b_{M-1}[n]}, \frac{\partial s[n]}{\partial a_1[n]}, \dots, \frac{\partial s[n]}{\partial a_N[n]} \right]^T.$$

Using (5), it is easy to get

$$\frac{\partial s[n]}{\partial b_i[n]} = x[n-i] + \sum_{k=1}^N a_k[n] \frac{\partial s[n-k]}{\partial b_i[n]}, \quad (10)$$

for $i = 0, 1, \dots, M-1$, and

$$\frac{\partial s[n]}{\partial a_i[n]} = s[n-i] + \sum_{k=1}^N a_k[n] \frac{\partial s[n-k]}{\partial a_i[n]}, \quad (11)$$

for $i = 1, \dots, N$. For simplification purpose, let us pose $\beta_i[n] = \partial s[n] / \partial b_i[n]$ and $\alpha_i[n] = \partial s[n] / \partial a_i[n]$. Moreover, assuming that the coefficients $b_i[n]$ and $a_i[n]$ vary slowly in time, as a usual simplification in the literature [3], we get

$$\frac{\partial s[n-k]}{\partial b_i[n]} \approx \frac{\partial s[n-k]}{\partial b_i[n-k]} = \beta_i[n-k], \quad (12)$$

and

$$\frac{\partial s[n-k]}{\partial a_i[n]} \approx \frac{\partial s[n-k]}{\partial a_i[n-k]} = \alpha_i[n-k], \quad (13)$$

for $k = 1, 2, \dots, N$. Now substituting (12) and (13) in (10) and (11), we obtain the following recursive equations:

$$\beta_i[n] = x[n-i] + \sum_{k=1}^N a_k[n] \beta_i[n-k], \quad (14)$$

and

$$\alpha_i[n] = s[n-i] + \sum_{k=1}^N a_k[n] \alpha_i[n-k]. \quad (15)$$

Let now us define the following vector

$$\boldsymbol{\eta}_n \in \mathbb{R}^{(M+N) \times 1} = [\beta_0[n], \beta_1[n], \dots, \beta_{M-1}[n], \alpha_1[n], \dots, \alpha_N[n]]^T, \quad (16)$$

hence the LMS algorithm, using (9) and (16), can be finally rewritten as

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_w[n] \varphi'_i(u_n) e[n] \boldsymbol{\eta}_n, \quad (17)$$

where the learning rate $\mu_w[n]$ absorbs the constants 2 and Δx .

For the adaptation of the spline control points, we proceed as shown in [9]

$$\Delta \mathbf{q}_{i,n} = \nabla_{\mathbf{q}_{i,n}} J(\mathbf{w}_n, \mathbf{q}_{i,n}) = -2e[n] \nabla_{\mathbf{q}_{i,n}} y[n] = -2e[n] \mathbf{C}^T \mathbf{u}_n, \quad (18)$$

obtaining the final algorithm

$$\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_q[n] e[n] \mathbf{C}^T \mathbf{u}_n, \quad (19)$$

where the learning rate $\mu_q[n]$ absorbs all constants.

From the computational point of view, in addition to the adaptation of the linear filter that, for the case of the IIR LMS algorithm, is equal to $2M + 4N$ multiplications plus $2M + 4N$ additions for each iteration, we have to consider the adaptation of the control points $\mathbf{q}_{i,n}$. In particular, for each iteration only the i th span of the curve is modified by calculating the quantities u_n , i and the expressions $\mathbf{u}_n^T \mathbf{C} \mathbf{q}_{i,n}$, $\dot{\mathbf{u}}_n^T \mathbf{C} \mathbf{q}_{i,n}$ and $\mathbf{C}^T \mathbf{u}_n$. Note that the calculation of the quantity $\mathbf{C} \mathbf{q}_{i,n}$ is executed during the output computation, as well as in the adaptation phase (the spline derivatives). 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. In any case, if $M + N \gg 4$, the computational overhead, for the nonlinear function computation and its adaptation, can be neglected with respect to the recursive linear filter.

3.1. Choice of the learning rate

The Taylor series expansion of the error $e[n+1]$ around the instant n , stopped at the first order is

$$e[n+1] = e[n] + \nabla_{\mathbf{w}_n}^T e[n] \Delta \mathbf{w}_n + \text{h.o.t.}, \quad (20)$$

where h.o.t. means high order terms. Now, using (17), we derive

$$\nabla_{\mathbf{w}_n}^T e[n] = -\varphi'_i(u_n) \dot{\mathbf{x}}_n^T, \quad (21)$$

$$\Delta \mathbf{w}_n = \mu_w[n] \varphi'_i(u_n) e[n] \boldsymbol{\eta}_n. \quad (22)$$

Substituting (21) and (22) in (20), we can obtain after simple manipulations

$$e[n+1] = \left[1 - \mu_w[n] \varphi_i'^2(u_n) \dot{\mathbf{x}}_n^T \boldsymbol{\eta}_n \right] e[n]. \quad (23)$$

Imposing for the convergence that $|e[n+1]| < |e[n]|$, the following constrain must be satisfied

$$\left| 1 - \mu_w[n] \varphi_i'^2(u_n) \dot{\mathbf{x}}_n^T \boldsymbol{\eta}_n \right| < 1 \quad (24)$$

which implies

$$0 < \mu_w[n] < \frac{2}{\varphi_i'^2(u_n) \dot{\mathbf{x}}_n^T \boldsymbol{\eta}_n}. \quad (25)$$

A similar constraint on $\mu_q[n]$ can be derived as done in [9].

4. Experimental results

A first experiment is performed in order to demonstrate the convergence behavior of the proposed IIR SAF. The experiment consists in the identification of an unknown Wiener system composed by a linear component, represented by the following recursive filter

$$H(z) = \frac{0.6 - 0.4z^{-1}}{1 + 0.2z^{-1} - 0.5z^{-2} + 0.1z^{-3}},$$

and a nonlinear memoryless target function implemented by a 23 points length LUT \mathbf{q}_0 , interpolated by a uniform third degree spline with an interval sampling $\Delta x = 0.2$ and defined as

$$\mathbf{q}_0 = \{-2.2, -2, -1.8, \dots, -1.0, -0.8, -0.91, 0.4, -0.2, 0.05, 0, -0.15, 0.58, 1.0, 1.0, 1.2, \dots, 2.0, 2.2\}.$$

Table 1

Mean values and variance of filter taps in first experiment averaged over 30 trials, for $a=0.5$ and B-spline basis.

Tap	\mathbf{w}_0	Mean	Variance $\times 10^{-5}$
b_0	0.60	0.611	3.240
b_1	-0.40	-0.404	7.455
a_1	0.20	0.201	11.506
a_2	-0.50	-0.500	6.659
a_3	0.10	0.100	10.523

Table 2

Mean values and variance of spline control points in first experiment averaged over 30 trials, for $a=0.5$ and B-spline basis.

i	\mathbf{q}_0	Mean	Variance $\times 10^{-5}$
7	-1.00	-0.965	9.254
8	-0.80	-0.828	1.920
9	-0.91	-0.878	5.105
10	-0.40	-0.395	2.018
11	-0.20	-0.195	6.850
12	0.05	0.056	7.416
13	0.00	-0.012	4.686
14	-0.15	-0.105	17.593
15	0.58	0.529	39.985
16	1.00	1.002	3.998

The input signal $x[n]$ consists in 30.000 samples of the signal generated by the following relationship:

$$x[n] = ax[n-1] + \sqrt{1-a^2}\xi[n], \tag{26}$$

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 between 0.1 and 0.95. In

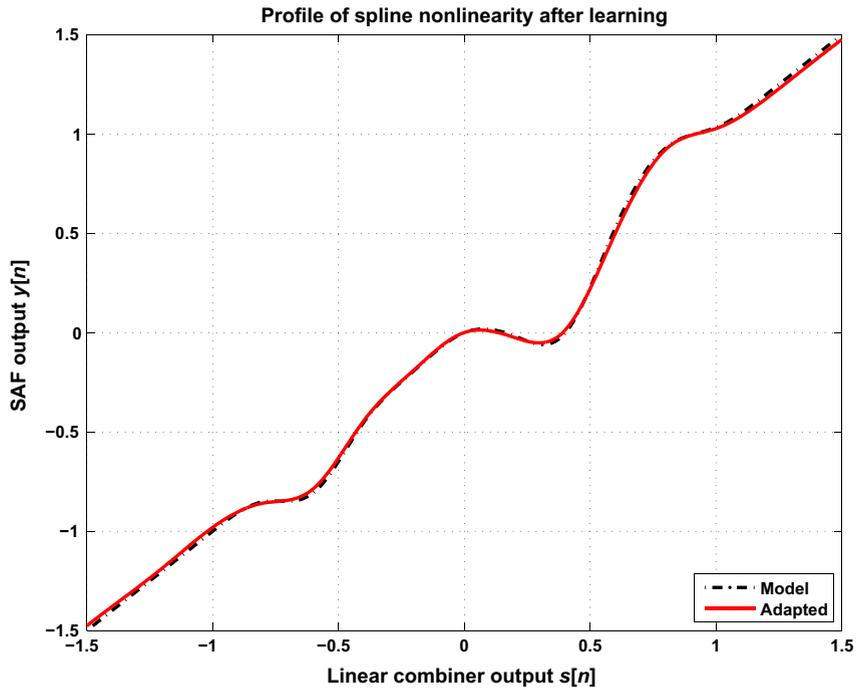


Fig. 2. Comparison of the model and adapted nonlinearity for first experiment, using (26) with $a=0.5$ and B-spline basis.

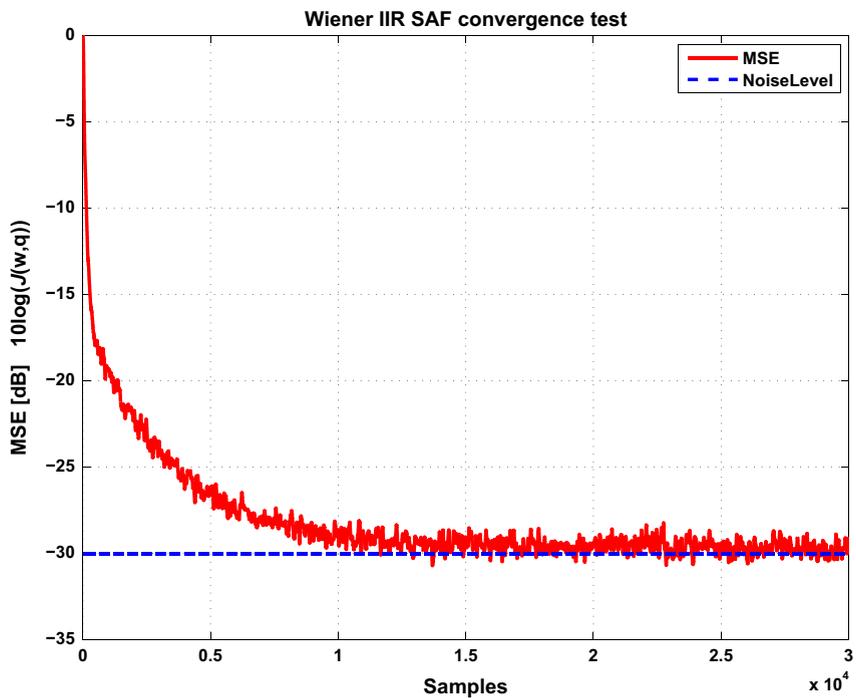


Fig. 3. MSE averaged over 30 trials of the proposed first experimental test, using (26) with $a=0.5$ and B-spline basis.

addition it is considered an additive white noise $v[n]$ with a signal to noise ratio $SNR=30$ dB. The learning rates are set to $\mu_w = \mu_q = 0.01$ and B-spline basis is used. The orders of MA and AR parts of the IIR adaptive filter are set to $M=2$ and $N=3$ respectively. We choose $\mathbf{w}_{-1} \in \mathbb{R}^{(M+N) \times 1} = [1, 0, \dots, 0]^T$ as initialization for filter weights, while spline control points \mathbf{q}_{-1} are initialized as a straight line with a unitary slope, conditions that give always good results in simulations. The choice of initializing the nonlinearity as a straight line is the simplest and most intuitively one. Since we have no *a priori* information on the shape of this nonlinearity, the line seems a good initial choice, i.e. at the beginning of the learning procedure the adaptive filter is linear. This means that if the model is also linear we have not to change the control points, while in the case mild nonlinearity the control points have not to move far from the line, on the contrary as they have made if we start from a random initialization or another shape.

Table 3

Summary of convergence test. The table shows the MSE for different values of the learning rate μ_w .

μ_w	MSE (dB)	μ_w	MSE (dB)
0.01	-29.6885	0.09	-12.1145
0.02	-28.9986	0.10	-11.7566
0.03	-28.6959	0.11	-10.4406
0.04	-28.4122	0.12	-9.6226
0.05	-27.7498	0.13	∞
0.06	-25.2897	0.14	∞
0.07	-24.9017	0.15	∞
0.08	-23.3056	0.16	∞

Results in the case of $a=0.5$, averaged over 30 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 central spline control points. Mean and variance are evaluated once the filter has completely converged. Moreover, Fig. 2 shows the profile of the spline function in the adaptive filter after the learning, while Fig. 3 shows the mean square error (MSE) over 30 trials. These figures clearly show the effectiveness of the proposed approach in identifying the given Wiener system.

In addition, in order to validate the bound in (25), we run simulation using different values of the learning rate μ_w chosen in the set $\{0.01, 0.02, \dots, 0.16\}$, while $\mu_q = 0.01$ and $a=0.9$. A summary of the MSE, averaged over 30 trials, is proposed in Table 3. The MSE value is obtained as a mean of the last 1000 samples of the squared error. This table shows that the architecture converges until $\mu_w = 0.12$ and then it diverges. We also evaluate the right side of (25) over 30 trials, obtaining a sequence of upper bounds for $\mu_w[n]$. The minimum of this sequence provides the value 0.1276, hence validating the bound in (25) and confirming the results in Table 3.

In a second experimental test, the system to identify is the Back and Tsoi NARMA model reported in [12]. This model consists in a cascade of the following 3-rd order IIR filter

$$H(z) = \frac{0.0154 + 0.0462z^{-1} + 0.0462z^{-2} + 0.0154z^{-3}}{1 - 1.99z^{-1} + 1.572z^{-2} - 0.4583z^{-3}}, \quad (27)$$

and the following nonlinearity

$$y[n] = \sin(s[n]). \quad (28)$$

The input signal $x[n]$ is the colored signal obtained with (26), choosing $a=0.95$ and consists of 5×10^4 samples. The

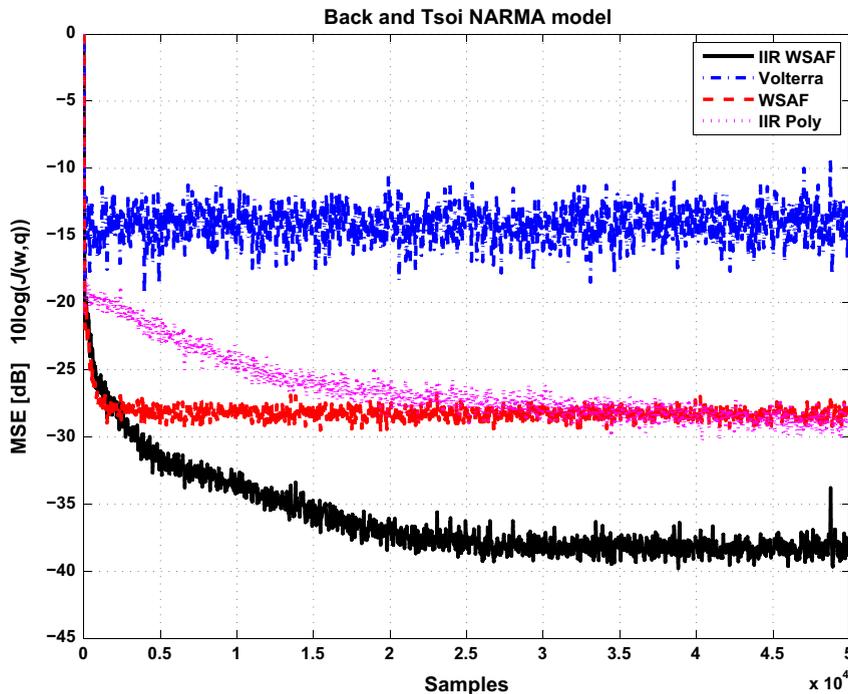


Fig. 4. MSE averaged over 30 trials of the proposed approach (IIR WSAF) for the Back and Tsoi NARMA model in (27) and (28) with $a=0.95$ and B-spline basis, compared with a full 3-rd order Volterra filter, a simple FIR SAF (WSAF) and a conventional IIR polynomial filter (IIR Poly).

learning rates are set to $\mu_w = \mu_q = 0.1$ and B-spline basis is used. The orders of MA and AR parts of the IIR adaptive filter are set to $M=4$ and $N=3$ respectively. Filter weights and spline control points are initialized as in the previous test.

Fig. 4 shows the MSE averaged over 30 trials of the IIR SAF approach compared with a full 3-rd order Volterra architecture with $M_v=15$ coefficients and adapted by a LMS algorithm with $\mu_v = 0.01$, a simple FIR SAF approach [9] using 15 filter taps $\mu_w = \mu_q = 0.02$ and a conventional IIR polynomial filter [4] using $\mu_p = 0.01$, $M=4$, $N=3$ and a 5-th order polynomial. The figure clearly demonstrates the superiority of the proposed approach even in the proposed case.

5. Conclusion

In this paper a variant of our previous SAF architecture [9] is proposed. In particular an IIR filter is used to implement the linear part of the proposed Wiener architecture. The new learning algorithm is derived based on a gradient descent approach. In addition a bound on the choice of the learning rate is also proposed.

Finally some experimental results on convergence and nonlinear Wiener system identification, prove the effectiveness of the proposed idea.

References

- [1] N. Wiener, *Nonlinear Problems in Random Theory*, MIT Press, Cambridge, MA, USA, 1958.
- [2] L. Ljung, *System Identification—Theory for the User*, 2nd edition, Prentice Hall, Upper Saddle River, NJ, 1999.

- [3] A. Uncini, *Fundamentals of Adaptive Signal Processing*, Springer-Verlag, Berlin, Germany, 2014.
- [4] V.J. Mathews, G.L. Sicuranza, *Polynomial Signal Processing*, John Wiley & Sons, Hoboken, New Jersey, 2000.
- [5] J.-M. Le Caillec, Spectral inversion of second order Volterra models based on the blind identification of Wiener models, *Signal Process.* 91 (11) (2011) 2541–2555.
- [6] R. Niemisto, T. Makela, V. Illyla, Robust fast affine projection algorithm for nonlinear acoustic echo cancellation, in: *Proceedings of European Signal Processing Conference (EUSIPCO)*, Toulouse, France, September 2002, pp. 523–526.
- [7] F. Albu, C. Kotropoulos, Modified Gauss-Seidel affine projection algorithm for acoustic echo cancellation, in: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, Philadelphia, PA, March 2005, pp. 121–124.
- [8] S. Haykin, *Neural Networks and Learning Machines*, 3rd edition, Prentice Hall, Upper Saddle River, NJ, 2009.
- [9] M. Scarpiniti, D. Comminiello, R. Parisi, A. Uncini, Nonlinear spline adaptive filtering, *Signal Process.* 93 (4) (2013) 772–783.
- [10] P.A. Regalia, *Adaptive IIR Filtering in Signal Processing and Control*, Marcel Dekker, New York, 1995.
- [11] A. Carini, V.J. Mathews, G.L. Sicuranza, Sufficient stability bounds for slowly varying direct-form recursive linear filters and their applications in adaptive IIR filters, *IEEE Trans. Signal Process.* 47 (9) (1999) 2561–2567.
- [12] A.D. Back, A.C. Tsoi, A simplified gradient algorithm for IIR synapse multilayer perceptrons, *Neural Comput.* 5 (3) (1993) 456–462.