A FAST RECURSIVE LEAST-SQUARES ADAPTIVE NONLINEAR FILTER*

V. John Mathews

Department of Electrical Engineering
University of Utah
Salt Lake City, Utah 84112

ABSTRACT

This paper presents a fast, recursive least-squares (RLS) adaptive nonlinear filter. The nonlinearity in the system is modeled using the Hammerstein model, which consists of a memoryless polynomial nonlinearity followed by a finite impulse response linear system. The complexity of our method is about $3NP^2+7NP+N+10P^2+6P$ multiplications per iteration and is substantially lower than the computational complexities of fast RLS algorithms that are direct extensions of RLS adaptive linear filters to the nonlinear case.

I. INTRODUCTION

Linear filtering of stationary and nonstationary signals has had enormous impact on the development of various techniques for processing such signals. While linear filters are inherently simple, there are several situation in which their performance is unacceptable. A very simple example is that of trying to relate two signals whose significant frequency components do not overlap.

In this paper we will present a fast, recursive least-squares (RLS) adaptive nonlinear filter. The nonlinearity employed in the structure is that of the Hammerstein model, which consists of a zero-memory nonlinearity followed by a linear system. We will further assume that the nonlinearity is mild enough so that it can be adequately represented using a low-order Taylor's series expansion. There are several applications for such "moderately" nonlinear filters. Due to imperfections in nonlinear amplifiers and companders and also due to possibly overdriven devices, nonlinear distortions are introduced into telephone networks. At high data rates, these distortions, even though only mildly nonlinear, will increase the bit error probability and therefore the channel distortions can no longer be modeled adequately as that due to linear system characteristics and additive noise. Several researchers have used nonlinear system representations for application in channel equalization, performance evaluation of data transmission systems [1,9,12], adaptive nonlinear noise cancellation [3,15] and a variety of other areas including process control [16] and device characterization [6,13].

Possibly because of their high computational complexity, very little work has been done in adaptively tracking time-varying nonlinear systems. Many of the past works employ Volterra series [14] representation of the nonlinear system and the least-mean-square (LMS) adaptation algorithm [3,7,8]. Computationally efficient gradient based adaptive nonlinear filters were studied in [4]. In many applications the slow convergence of the LMS algorithm is not acceptable. A block adaptive nonlinear filter employing an iterative least-squares solution was introduced in [15]. The method presented here is a continuously adaptive RLS approach. Continuously adaptive (RLS) second order Volterra filters were studied in [5,17]. However, both the methods assume the structure derived for Gaussian input signals and consequently do not work very well when there are derivations from the Gaussian signal assumption. This paper's method presents an exact solution to the least-squares problem and therefore will work well with any type of input signals. Furthermore, we will see that the method presented here makes efficient use of the structure of the nonlinearity and, as a result, is considerably simpler than direct extensions of the fast RLS algorithm to the nonlinear case.

II. PROBLEM STATEMENT AND THE FAST, RLS ADAPTIVE NONLINEAR FILTER

Let $d(n)$ and $x(n)$ represent the reference and primary inputs, respectively, to the adaptive filter. Then, the problem considered in the paper is that of finding an exponentially windowed, fast RLS solution for the linear and nonlinear coefficients of the adaptive filter that minimizes the cost function

$$J(n) = \sum_{k=0}^{n} \lambda^{n-k} \left(d(k) - \hat{d}_n(k)\right)^2 \qquad (1)$$

at each time instant n. In Eq. 1, the estimate $\hat{d}_n(k)$ is obtained as the output of a linear, finite impulse response filter whose input sequence is a nonlinear function $f_n(\cdot)$ of the primary input sequence $x(n)$. That is,

$$\hat{d}_n(k) = \sum_{i=0}^{N-1} h_i(n)\, f_n\left(x(k-i)\right) \qquad (2)$$

*This research was supported in part by the National Science Foundation under Grant #MIP-8708970 and by a University of Utah research grant.

21ACSSC-1/88/0156 $01.00 © 1988 MAPLE PRESS

Also, $0 < \lambda \leq 1$ is the parameter of the exponential window that controls the rate at which the adaptive filter tracks time-varying parameters and $f_n(x)$ is a static nonlinearity that can be adequately modeled using a P-th order Taylor series expansion

$$f_n(x) = \sum_{j=1}^{P} a_j(n)\, x^j . \tag{3}$$

Now the problem becomes one of estimating the linear coefficients $h_i(n)$ and nonlinear coefficients $a_j(n)$ such that Eq. 1 is minimized.

The nonlinear structure that is described by Eq. 2 and 3 is known as the Hammerstein model and a block diagram for the resulting adaptive filter structure is shown in Fig. 1. Note that Eq. 2 and 3 can be combined to get a linear representation for $d_n(k)$ in various power of $x(k-i)$ as

$$\hat{d}_n(k) = \sum_{i=0}^{N-1} \sum_{j=1}^{P} w_{i,j}(n)\, x^j(k-i) \tag{4}$$

where

$$w_{ij}(n) = h_i(n)\, a_j(n); \quad 0 \leq i \leq N-1, \quad 1 \leq j \leq P . \tag{5}$$

So as to be able to uniquely solve for $h_i(n)$ and $a_j(n)$, we will constrain

$$a_1(n) = 1 \quad \text{for all } n. \tag{6}$$

With this constraint,

$$h_i(n) = w_{i1}(n) \quad ; \; 0 \leq i \leq N-1 \tag{7}$$

and

$$a_j(n) = \frac{1}{S(n)} \sum_{i=0}^{N-1} w_{ij}(n) \quad ; \; 1 \leq j \leq P \tag{8}$$

where

$$S(n) = \sum_{i=0}^{N-1} h_i(n). \tag{9}$$

We will also assume that $S(n) \neq 0$ except at those times when $h_i(n) = 0$ for $0 \leq i \leq N-1$. While these constraints exclude certain types of systems (for example, note that Eq. 6 implies that the nonlinear function $f_n(x)$ must have a linear component $a_1(n)\, x$), there is a large class of systems for which the algorithms developed in this paper will work well.

To derive the fast, RLS adaptive nonlinear filter, we make use of the fast algorithms in [2,11] and apply it to our structure. Because of stability considerations, the algorithm that is presented here is not the simplest possible one. However, experimental results have indicated that the extra computational complexity does considerably improve the numerical stability of the algorithm.

The fast RLS, nonlinear adaptive filter makes use of four nonlinear filters (with the same structure as in Fig. 1) in parallel. Different parameters of these filters and other associated variables that appear in the algorithm are defined in Table 1.

The key step that provides simplifications in our algorithm is the structure of the solution of Eq. 5 given in Eq. 7 and 8. Let $\tilde{a}_j(n) = S(n)\, a_j(n)$. Substituting this and Eq. 6 in Eq. 4, we can see that

$$\hat{d}(k) = \frac{1}{\tilde{a}_1(n)} \sum_{i=0}^{N-1} h_i(n) \sum_{j=1}^{P} \tilde{a}_j(n)\, x^j(k-i) . \tag{10}$$

Further, let us define vectors $w(n)$ and $k_N(n)$ of size NP such that

$$W(n) = \left[w_{0,1}(n), w_{0,2}(n), \ldots, w_{0,P}(n) \ldots, w_{1,1}(n), \ldots, w_{1,P}(n) \ldots, w_{N-1,1}(n) \ldots, w_{N-1,P}(n)\right]^T \tag{11}$$

and

$$k_N(n) = \left[h_{C,N,0}(n) \cdot a_{C,N,1}(n),\, h_{C,N,0}(n) \cdot a_{C,n,2}(n), \ldots, h_{C,N,N-1}(n) \cdot a_{C,N,P}(n)\right]^T . \tag{12}$$

These are just expanded vectors obtained from the linear and nonlinear coefficient vectors and when they act on an appropriately defined input vector will behave in a manner that is equivalent to the filter structure in this paper.

It is well known [2,11] that the W(n) vector can be updated in fast RLS algorithms using the equation

$$W(n) = W(n-1) + k_N(n)\, \varepsilon_N(n) \tag{13}$$

where $\varepsilon_N(n)$ is the predicted estimation error and is defined in Table 1. Applying Eq. 7 and 8 to 13 and using the scaled nonlinear coefficients ($\tilde{a}_j(n)$, $\tilde{a}_{C,N,j}(n)$, etc.), we get the update equations for the linear and nonlinear coefficient vectors as

$$\tilde{A}(n) = \tilde{A}(n-1) + \tilde{A}_{C,N}(n)\, \varepsilon_N(n) \tag{14}$$

and

$$H(n) = H(n-1) + H_{C,N}(n)\, \varepsilon_N(n) \tag{15}$$

The derivation of the rest of the update equations is similar and will be omitted here. The complete algorithm is described in Table 2.

Remarks: 1. The algorithm was initialized using zero values for all filter coefficients and the associated parameters except

$$\alpha_N(-1) = \mu\, I_{PxP} \tag{16}$$

Exact initialization is possible, but we did not use it for the sake of numerical stability.

2. An operations count will show that our algorithm requires $3NP^2+7NP+N+6P+10P^2$ multiplications and $4P + 2$ divisions per data sample. This represents a substantial reduction in the computational complexity from those methods that are direct extensions of fast, transversal multichannel filters to the nonlinear case.

III. CONCLUSIONS

This paper presented a fast recursive least-squares adaptive nonlinear filter. The nonlinearity used in the structure was the Hammerstein model. By making use of the structure of the nonlinearity, the algorithm reduces the computational complexity of direct extensions of fast RLS adaptive linear filters by more than half when N is somewhat larger than P. This saving in computational complexity is substantial and it is expected that the algorithm will find applications in several practical problems.

Experimental results have indicated that our algorithm works well in many situations. However, the aspects of numerical stability of the algorithm need further study. Several techniques available for improving the numerical stability of fast RLS algorithms were tested and they do seem to improve the numerical properties of the fast RLS nonlinear filter. We are in the process of studying several other structures that have better numerical stability than the present algorithm. Several simplifications to the algorithm presented in the paper using techniques employed in [2] are also being studied now.

Acknowledgment. The author thanks Junghsi Lee for several useful discussion on the contents of this paper.

REFERENCES

1. S. Benedetto, E. Biglieri and R. Daffara, "Performance of multilevel baseband digital systems in a nonlinear environment," IEEE Transactions on Communications, Vol. COM-24, No. 10, pp. 1166-1175, October 1976.
2. J. M. Cioffi and T. Kailath, "Fast, recursive least-squares transversal filters for adaptive filtering," IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. ASSP-32, No. 2, pp. 304-337, April 1984.
3. M. J. Coker and D. N. Simkins, "A nonlinear adaptive noise canceller," Proceedings 1980 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 470-473, 1980.
4. C. F. N. Cowan and P. F. Adams, "Nonlinear system modelling: Concept and application," Proc. ICASSP, pp. 45.6.1-4, San Diego, California, April 1984.
5. C. E. Davila, A. J. Welch and H. G. Rylander, III, "A second-order adaptive Volterra filter with rapid convergence," IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. ASSP-35, No. 9, pp. 1259-1263, September 1987.
6. A. Javed, B. A. Syrett and P. A. Goud, "Intermodulation distortion analysis of reflection-type IMPATT amplifiers using Volterra series representation," IEEE Transactions on Microwave Theory and Techniques,Vol. MTT-25, No. 9, pp. 729-733, September 1977.
7. T. Koh and E. J. Powers, "Second order Volterra filtering and its application to nonlinear system identification," IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. ASSP-33, No. 6, pp. 1445-1455, December 1985.
8. T. Koh and E. J. Powers, "An adaptive nonlinear digital filter with lattice orthogonalization," Proceedings of ICASSP 1983, pp. 37-40, Boston, 1983.
9. T. L. Lim and J. K. Omura, "Error rate estimates in digital communication over a nonlinear channel with memory," IEEE Transactions on Communications, Vol. COM-31, No. 3, pp. 407-412, March 1983.
10. D. W. Lin, "On digital implementation of the fast Kalman algorithm," IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. ASSP-32, No. 5, pp. 998-1005, October 1984.
11. L. Ljung, M. Morf and D. Falconer, "Fast calculation of gain matrices for recursive estimation schemes," International Journal of Control, Vol. 27, pp. 1-19, January 1978.
12. M. Maqusi, "Performance of baseband digital data transmission in nonlinear channels with memory," IEEE Transactions on Communications, Vol. COM-33, No. 7, pp. 715-719, July 1985.
13. W. Reiss, "Nonlinear distrotion analysis of P-I-N diode attenuators using Volterra series representation," IEEE Transactions on Circuits and Systems, Vol. CAS-31, No. 6, pp. 535-542, June 1984.
14. M. Schetzen, The Volterra and Wiener Theory of the Nonlinear Systems, Wiley, New York, 1980.
15. J. C. Stapleton and S. C. Bass, "Adaptive noise cancellation for a class of nonlinear, dynamic reference channels," IEEE Transactions on Circuits and Systems, Vol. CAS-32, No. 2, pp. 143-150, February 1985.
16. R. Takahashi and A. Arakawa, "Real-time application of simultaneous estimation theory for parameters and state to one-dimensional heat conduction," Nuclear Engineering and Design, Vol. 81, No. 3, pp. 437-449, 1984.
17. D. H. Youn, K. K. Yu, and V. J. Mathews, "Adaptive nonlinear digital filter with sequential regression algorithm, "Proc. 22nd Annual Allerton Conference on Control Communication and Computing," pp. 152-161, Urbana, Illinois, October 3-5, 1984.

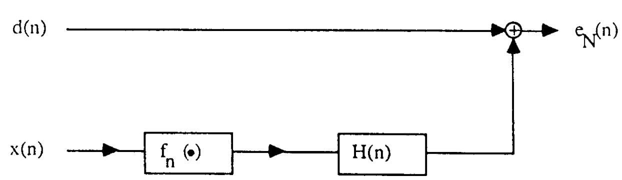

Fig. 1. Structure of the nonlinear filter in this paper.

Table 1. Fast, RLS Adaptive Nonlinear Filter Variables

Primary input: $x(n)$

$$X(n) = [x(n), x^2(n), \ldots, x^P(n)]^T$$

$$X_N(n) = [X(n), X(n-1), \ldots, X(n-N+1)]$$

Reference input: $d(n)$

	Variable	Definition	Remarks
I.	Filter		
	$H(n)$	$[h_0(n), h_1(n), \ldots, h_{N-1}(n)]^T$	Linear coefficients
†	$\tilde{A}(n)$	$[\tilde{a}_1(n), \tilde{a}_2(n), \ldots, \tilde{a}_P(n)]^T$	Scaled nonlinear coefficients
	$\varepsilon_N(n)$	$d(n) - \frac{1}{\tilde{a}_1(n-1)} [\tilde{A}_N^T(n-1)\, X_N(n)]\; H_N(n-1)$	Predicted estimation error
	$e_N(n)$	$d(n) - \frac{1}{\tilde{a}_1(n)} [\tilde{A}_N^T(n)\, X_N(n)]\; H_N(n)$	Error
II.	Forward Predictor		
††	$H_F(n)$	$[H_{f,1}(n), H_{f,2}(n), \ldots, H_{f,P}(n)]$	Linear coefficient matrix
	$\tilde{A}_F(n)$	$[\tilde{A}_{f,1}(n), \tilde{A}_{f,2}(n), \ldots, \tilde{A}_{f,P}(n)]$	Scaled nonlinear coefficient matrix
	$\eta_N(n)$	$[\eta_{N,1}(n), \eta_{N,2}(n), \ldots, \eta_{N,P}(n)]^T$	Predicted forward prediction errors*
	$f_N(n)$	$[f_{N,1}(n), f_{N,2}(n), \ldots, f_{N,P}(n)]^T$	Forward prediction errors*
	$\alpha_N(n)$	$\sum_{k=0}^{n} \lambda^{n-k}\; f_N(k)\, f_N^T(k)$	Forward prediction error power
III.	Backward Predictor		
	$H_B(n)$	$[H_{b,1}(n), H_{b,2}(n), \ldots, H_{b,P}(n)]$	Linear coefficient matrix
	$\tilde{A}_B(n)$	$[\tilde{A}_{b,1}(n), \tilde{A}_{b,2}(n), \ldots, \tilde{A}_{b,P}(n)]$	Scaled nonlinear coefficient matrix
	$\psi_N(n)$	$[\psi_{N,1}(n), \psi_{N,2}(n), \ldots, \psi_{N,P}(n)]$	Predicted backward prediction errors*
IV.	Gain Vectors		
	$H_{C,N}(n)$	$[h_{C,N,0}(n), h_{C,N,2}(n), \ldots, h_{C,N,N-1}(n)]^T$	Linear coefficients
	$\tilde{A}_{C,N}(n)$	$[\tilde{a}_{C,N,1}(n), \tilde{a}_{C,N,2}(n), \ldots, \tilde{a}_{C,N,P}(n)]^T$	Scaled nonlinear coefficients

† Quantities with ~ are scaled by the sum of the corresponding linear coefficients.

†† If A is an LxM matrix, A_j denotes the j-th column of A and $a_{j,k}$ denotes the k-th element of the j-th column of A.

* See Table 2 for definition of prediction errors $\eta_{N,j}(n)$, $f_{N,j}(n)$, and $\psi_{N,j}(n)$.

Table 2. Fast, RLS, Adaptive Nonlinear Filter.

$$\eta_{N,j}(n) = x^j(n) + \frac{1}{\tilde{a}_{f,j,1}(n-1)} \left[\tilde{A}^T_{f,j}(n-1)\ X_N(n-1)\right]\ H_{f,j}(n-1);\ j = 1,2,\ldots,P \tag{2.1}$$

$$\tilde{A}_F(n) = \tilde{A}_F(n-1) - \tilde{A}_{C,N}(n-1)\ \eta^T_N(n) \tag{2.2}$$

$$H_F(n) = H_F(n-1) - H_{C,N}(n-1)\ \eta^T_N(n) \tag{2.3}$$

$$f_{N,j}(n) = x^j(n) + \frac{1}{\tilde{a}_{f,j,1}(n)} \left[\tilde{A}^T_{f,j}(n)\ X_N(n-1)\right]\ H_{f,j}(n);\ j = 1,2,\ldots,P. \tag{2.4}$$

$$\alpha_N^{-1}(n) = \frac{1}{\lambda}\left[\alpha_N^{-1}(n-1) - \frac{\alpha_N^{-1}(n-1)\ \eta_N(n)\ f^T_N(n)\ \alpha_N^{-1}(n-1)}{\lambda + f^T_N(n)\ \alpha_N^{-1}(n-1)\ \eta_N(n)}\right];\ j = 1,2,\ldots,P. \tag{2.5}$$

$$\sigma(n) = \alpha_N^{-1}(n)\ f_N(n) \tag{2.6}$$

$$\tilde{A}_{C,N+1}(n) = \tilde{A}_{C,N}(n-1) + [\tilde{A}_F(n) + I_{PxP}]\ \ \sigma(n) \tag{2.7}$$

$$H_{C,N+1}(n) = \begin{pmatrix} 0 \\ \hline H_{C,N}(n-1) \end{pmatrix} + \begin{pmatrix} \sigma_1(n) \\ \hline H_F(n)\ \bar{\sigma}(n) \end{pmatrix} \tag{2.8}$$

$$\psi_{N,j}(n) = x^j(n-N) + \frac{1}{\tilde{a}_{b,j,1}(n-1)} \left[\tilde{A}^T_{b,j}(n-1)\ X_N(n)\right]\ H_{b,j}(n-1) \tag{2.9}$$

$$\tau(n) = \tilde{A}_{C,N+1}(n) \left[h_{C,N+1,N}(n) \Big/ \tilde{a}_{C,N+1,1}(n)\right] \tag{2.10}$$

$$\tilde{A}_t(n) = \tilde{A}_{C,N+1}(n) - \tau(n) \tag{2.11}$$

$$\begin{pmatrix} H_t(n) \\ \hline h_{C,N+1,N}(n) \end{pmatrix} = H_{C,N+1}(n) \tag{2.12}$$

$$\tilde{A}_B(n) = \left[\tilde{A}_B(n-1) - \tilde{A}_{t,N}(n)\ \psi^T_N(n)\right]\ \left(I_{PxP} - \tau(n)\ \psi^T_N(n)\right)^{-1} \tag{2.13}$$

$$H_B(n) = \left[H_B(n-1) - H_{t,N}(n)\ \psi^T_N(n)\right]\ \left(I_{PxP} - \tau(n)\ \psi^T_N(n)\right)^{-1} \tag{2.14}$$

$$\tilde{A}_{C,N}(n) = \tilde{A}_t(n) - \tilde{A}_B(n)\ \tau(n) \tag{2.15}$$

$$H_{C,N}(n) = H_t(n) - H_B(n)\ \tau(n) \tag{2.16}$$

$$\varepsilon_N(n) = d(n) - \frac{1}{\tilde{a}_1(n-1)} \left[\tilde{A}^T\ (n-1)\ X_N(n)\right] H(n-1) \tag{2.17}$$

$$\tilde{A}(n) = \tilde{A}(n-1) + \tilde{A}_{C,N}(n)\ \varepsilon_N(n) \tag{2.18}$$

† $$A(n) = \frac{1}{\tilde{a}_1(n)}\ \tilde{A}(n) \tag{2.19}$$

† $$e_N(n) = d(n) - \frac{1}{\tilde{a}_1(n)} \left[\tilde{A}^T(n)\ X_N(n)\right] H(n) \tag{2.20}$$

† Compute only if necessary