Output-Error Adaptive Bilinear Filters

Junghsi Lee and V. John Mathews

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

Abstract

This paper presents an overview of several gradient-type recursive algorithms for adaptive nonlinear filters equipped with bilinear system models. Bilinear models are attractive because they can approximate a large class of nonlinear systems with great parsimony in the use of coefficients. Two algorithms of complexity $O(N^3)$ (N is the memory span of the bilinear system model used in the adaptive filter) multiplications per sample and two other algorithms of complexity $O(N^2)$ multiplications per sample are presented in this paper. The results of several computer experiments show that at least one of the $O(N^2)$ complexity algorithms works almost as well as the more complex algorithms.

1 Introduction

This paper introduces and compares several adaptive nonlinear filters that employ the bilinear system model. The input-output relationship for a bilinear system is given by

$$y(n) = \sum_{i=0}^{N-1} a_i x(n-i) + \sum_{i=1}^{N-1} b_i y(n-i) + \sum_{i=0}^{N-1}\sum_{j=1}^{N-1} c_{i,j} x(n-i)y(n-j) \qquad (1)$$

where x(n) and y(n) are the input and output signals, respectively, of the system. This model is attractive because it can adequately approximate a large class of nonlinear systems with good parsimony in the use of coefficients, much like linear infinite-impulse-response (IIR) filters can approximate a large class of linear systems with good parsimony. Bilinear system models have been used in a wide class of problems in engineering, biology, socio-economics, and ecology [2], [7].

In spite of the potential benefits of bilinear system models, the theory of adaptive nonlinear filters employing nonlinear feedback structures is still very much in its infancy. Some very recent work in adaptive bilinear filtering employ recursive least-squares (RLS) adaptation algorithms or its variations [1], [5]. Among the techniques that are available, even the most computationally efficient schemes require $O(N^3)$ multiplications per sample. It is the purpose of this paper to introduce several LMS type adaptive bilinear filters. Among these filters are two simplified structures which need $O(N^2)$ multiplications per sample and still performs reasonably well.

The remainder of this paper is organized as follows. The following section describes several LMS-type adaptive bilinear filtering algorithms. Section III contains simulation results that demonstrate and compare the performances of the algorithms. Finally, the concluding remarks are made in Section IV.

2 Derivations of the Adaptive Bilinear Filters

We will first derive an output-error LMS-type adaptive bilinear filtering algorithm. For convenience, let us denote it as ABF1. This algorithm has a computational complexity of N^3+3N^2-N. We also present a simplified version (ABF2) of the ABF1. ABF2 requires $5N^2-N$ computations per sample. A further simplified version (AFB3) that needs $2N^2+2N$ computations per sample only will then be introduced. Finally, we will present bilinear filtering algorithm (ABF4) which is the extension to Fan's work [3]. The computational complexity of ABF4 is the same as that of ABF1. It must be noted that all the algorithms are extensions of adaptive IIR filters available for linear system models to the nonlinear case. Our objective is to study the usefulness of such techniques in adaptive nonlinear filtering problems.

2.1 Derivation of ABF1

Let d(n) and x(n) represent the desired response signal and the input signal, respectively, to the adaptive filter. Let $\hat{d}(n)$ denote the *a priori* output of the adaptive filter at time n, i.e.,

$$\hat{d}(n) = \sum_{i=0}^{N-1} \hat{a}_i(n-1)x(n-i) + \sum_{i=1}^{N-1} \hat{b}_i(n-1)\hat{d}(n-i) + \sum_{i=0}^{N-1}\sum_{j=1}^{N-1} \hat{c}_{i,j}(n-1)x(n-i)\hat{d}(n-j) \quad (2)$$

In (2), $\hat{a}_i(n-1)$, $\hat{b}_i(n-1)$ and $\hat{c}_{i,j}(n-1)$ are the coefficients at time n-1 of the adaptive bilinear filter.

For simplicity, vector representation will be employed. The input vector $\hat{X}(n)$ at time n, which has N^2+N-1 elements, is defined as

$$\begin{aligned}\hat{X}(n) = [&x(n), x(n)\hat{d}(n-1), ..., x(n)\hat{d}(n-N+1), \hat{d}(n-1),\\ &x(n-1)\hat{d}(n-1), ..., x(n-N+1)\hat{d}(n-1), ...,\\ &x(n-N+2), x(n-N+2)\hat{d}(n-N+1), \hat{d}(n-N+1),\\ &x(n-N+1)\hat{d}(n-N+1), x(n-N+1)\,]^T, \end{aligned} \quad (3)$$

where the superscript T denotes transposition. Also, the N^2+2N-1 coefficient vector W_n at time n is defined as

$$\begin{aligned}W_n = [&\hat{a}_0(n), \hat{c}_{0,1}(n), ..., \hat{c}_{0,N-1}(n), \hat{b}_1(n), \hat{c}_{1,1}(n), ..., \hat{c}_{N-1,1}(n)\\ &..., \hat{a}_{N-2}(n), \hat{c}_{N-2,N-1}(n), \hat{b}_{N-1}(n), \hat{c}_{N-1,N-1}(n), \hat{a}_{N-1}(n)\,]^T \end{aligned} \quad (4)$$

The problem is then to find a gradient descent solution for the coefficients of the adaptive filter which attempts to minimize the cost function

$$\begin{aligned}J(n) &= (d(n) - \hat{d}(n))^2\\ &= (d(n) - W_{n-1}^T\hat{X}(n))^2 \end{aligned} \quad (5)$$

at each time. The gradient descent solution is given by

$$W_n = W_{n-1} - \frac{\mu}{2}\frac{\partial(d(n) - W_{n-1}^T\hat{X}(n))^2}{\partial W_{n-1}} \quad (6)$$

where μ is a small positive constant that controls the rate at which the adaptive filter converges. Note that $\hat{d}(n)$ and therefore $\hat{X}(n)$ are functions of W_{n-1}. The gradient in (6) may be obtained from

$$-\frac{\partial(d(n) - W_{n-1}^T\hat{X}(n))}{\partial W_{n-1}} = \hat{X}(n) + \sum_{i=1}^{N-1}\hat{b}_i(n-1)\frac{\partial\hat{d}(n-i)}{\partial W_{n-1}} + \sum_{i=0}^{N-1}\sum_{j=1}^{N-1}\hat{c}_{i,j}(n-1)x(n-i)\frac{\partial\hat{d}(n-j)}{\partial W_{n-1}} \quad (7)$$

Note that (7) indicates the necessity of reevaluation of the derivatives of past values of $\hat{d}$ with respect to W_{n-1}. An assumption commonly employed in the adaptive IIR filtering literature is that μ is sufficiently small such that $W_{n-1} \approx W_{n-2} \approx \ldots \approx W_{n-N}$ [6]. Using this approximation and the fact that d(n) is not a function of W_{n-1}, we may write (7) as

$$\frac{\partial\hat{d}(n)}{\partial W_{n-1}} = \hat{X}(n) + \sum_{i=1}^{N-1}\hat{b}_i(n-1)\frac{\partial\hat{d}(n-i)}{\partial W_{n-1-i}} + \sum_{i=0}^{N-1}\sum_{j=1}^{N-1}\hat{c}_{i,j}(n-1)x(n-i)\frac{\partial\hat{d}(n-j)}{\partial W_{n-1-j}} \quad (8)$$

Let

$$\psi(n) = \frac{\partial\hat{d}(n)}{\partial W_{n-1}} \quad (9)$$

Then, (8) simplifies to

$$\psi(n) = \hat{X}(n) + \sum_{i=1}^{N-1}\hat{b}_i(n-1)\psi(n-i) + \sum_{i=0}^{N-1}\sum_{j=1}^{N-1}\hat{c}_{i,j}(n-1)x(n-i)\psi(n-j) \quad (10)$$

Finally, we summarizes ABF1 as follows.

$$\varepsilon(n) = d(n) - W_{n-1}^T\hat{X}(n) \quad (11)$$

$$\psi(n) = \hat{X}(n) + \sum_{j=1}^{N-1}\psi(n-j)\{\hat{b}_j(n-1) + \sum_{i=0}^{N-1}\hat{c}_{i,j}(n-1)x(n-i)\} \quad (12)$$

$$W_n = W_{n-1} + \mu\varepsilon(n)\psi(n) \quad (13)$$

The above algorithm is an adaptation of [6] to the bilinear case. As discussed earlier, the computational complexity of ABF1 is N^3+3N^2-N multiplications per sample and it needs to store N-1 past values of the N^2+N-1 element vector ψ. By employing arguments similar to those used in [6], the computational burden of the adaptive filter can be reduced. This will result in algorithm ABF2.

2.2 Derivation of ABF2

The key to the derivation of ABF2 is the fact that the last N^2-N-1 elements of $\hat{X}(n)$ in (3) are delayed versions of

the first 2N elements of $\hat{X}(n)$. This motivates the approximation of replacing the last N^2-N-1 elements of ψ with appropriate delayed versions of the first 2N elements of ψ. Instead of (12), ABF2 uses

$$\psi_{2N}(n) = \hat{X}_{2N}(n) + \sum_{j=1}^{N-1} \psi_{2N}(n-j) \{ \hat{b}_j(n-1) + \sum_{i=0}^{N-1} \hat{c}_{i,j}(n-1)x(n-i) \}, \quad (14)$$

where ψ_{2N} and $\hat{X}_{2N}$ denote the vectors that contain the first 2N entries of ψ and $\hat{X}$, respectively. The N^2+N-1 element vector ψ, which is needed for (13), is constructed from the current and past values of ψ_{2N}. Note that ABF2 needs to store N-1 past values of the 2N element vector ψ_{2N} and requires only $5N^2$–N multiplications per sample.

2.3 Derivation of ABF3

In 1976, Feintuch proposed a highly simplified linear adaptive recursive LMS filtering algorithm [4]. He proposed to use

$$\psi(n) = \hat{X}(n) \quad (15)$$

for the gradient vector. This can be viewed as a simplified version obtained by neglecting the second term on the right hand side of (12). Adaptation of this idea will result in an adaptive bilinear filter that needs only $2N^2$+2N multiplications per sample.

2.4 Derivation of ABF4

Recently, Fan proposed some adaptive IIR filters [3] which are based on the Steiglitz-McBride scheme. We have also extended Fan's filters to adaptive bilinear filters. The resulting algorithms are similar to ABF1 and ABF2, depending on whether ψ or ψ_{2N} is employed. We will discuss the scheme (ABF4) involving ψ here. The only difference be-tween ABF1 and ABF4 is that the latter uses d, in comparison with $\hat{d}$ used by ABF1, in (12). That is

$$\psi(n) = X(n) + \sum_{j=1}^{N-1} \psi(n-j) \{ \hat{b}_j(n-1) + \sum_{i=0}^{N-1} \hat{c}_{i,j}(n-1)x(n-i) \}, \quad (16)$$

where X(n) is defined in a similar way as $\hat{X}(n)$ with the exception that d is used in place of $\hat{d}$ in the definition.

3 Simulation Results

In this section, we present the results of several experiments. The problem considered here is that of identifying a bilinear system with N=3. The coefficients of the unknown bilinear system can be found in Table I. The input signal x(n) was a white, zero-mean and pseudorandom Gaussian noise. The variance of x(n) was 0.18, and the power of y(n) was about unity. We first considered a white, Gaussian, zero-mean measurement noise with variance ξ_o. Two different noise levels were studied: ξ_o=0.001 and ξ_o=0.1. We also conducted the experiments where the measurement noise was colored. The colored noise was obtained through the equation

$$v(n) = \eta(n) - 0.1v(n) + 0.8v(n-2), \quad (17)$$

where $\eta(n)$ belonged to a zero-mean white Gaussian process. The variance of $\eta(n)$ was adjusted such that the variance of v(n) (for convenience, denote it as ξ_o again) was either 0.1 or 0.001 approximately. The adaptive filter was implemented with the same structure and the same number of coefficients as that of the unknown system. The step-size μ was set to 0.009. All of the results presented were obtained from ensemble averages over fifty independent runs.

Tables II and III contain the mean and the variance values of selected filter coefficients when the measurement noises were white and colored, respectively. These values were obtained by time averaging the corresponding ensemble averaged sequences in the range [9000,10000]. The steady-state *a priori* mean-squared error values obtained in a similar manner are given in Table IV. All the algorithms seem to work reasonably well. It appears that the performances of ABF1 and ABF2 are very similar even though ABF2 is an order of magnitude less complex than ABF1. ABF3 also works quite well when the measurement noise is white. The coefficient trajectories, which are not presented here, indicated that AFB4 has a slower convergence rate than the other algorithms. Also, AFB4 does not work well when the measurement noise is colored and the noise level is relatively high.

4 Concluding Remarks

This paper presented an overview of several LMS type adaptive bilinear filtering algorithms. The results of a large number of experiments showed that the algorithms work well in several situations. While it is too early to draw definitive conclusions, it is possible that the more efficient algorithms perform as well as the others in adaptive identification of bilinear systems. On the basis

of the experiments presented, it appears that ABF2, because of its simplicity and good performance, may be a compromise choice in many applications. However, much more analytical and empirical works need to be done to further evaluate the performance of these algorithms.

One significant problem that we did not address in this paper is that of the stability of the identified systems. In general, the stability conditions for bilinear systems depends on the characteristics of the input signal also, and this complicates the problem. While there are several works on the stability of bilinear systems [8], [9] available in the literature, the on-line implementation of such schemes to monitor the stability of the systems is very computationally intense. We are currently investigating simplified algorithms for real-time stability monitoring.

Acknowledgement

This work was supported in part by NSF Grant No. MIP 8922146.

References

[1] H. K. Baik and V. J. Mathews, "Adaptive Lattice Bilinear Filters," submitted to *IEEE Transactions on Acoustics, Speech, and Signal Processing*, December 1990.

[2] C. Bruni, G. Dipillo and G. Koch, "Bilinear Systems: An Appealing Class of 'Nearly Linear' Systems in Theory and Applications,"*IEEE Transactions on Automatic Control*, Vol. AC-19, No. 4, pp. 334-348, August 1974.

[3] H. Fan and W. K. Jenkins, "A New Adaptive IIR Filter," *IEEE Transactions on Circuits and Systems*, Vol. CAS-33, No. 10, pp. 939-947, October 1986.

[4] P. L. Feintuch, "An Adaptive Recursive LMS Filter," *Proceedings of the IEEE*, Vol. 64, pp. 1622-1624, November 1976.

[5] F. Fnaiech and L. Ljung," Recursive Identification of Bilinear Systems," *International Journal of Control*, Vol. 45, No. 2, pp. 453-470, 1987.

[6] C. R. Johnson, Jr., "Adaptive IIR Filtering: Current Results and Open Issues," *IEEE Transactions on Information Theory*, Vol. IT-30, No. 2, pp. 237-250, March 1984.

[7] R. R. Mohler, *Bilinear Control Processes*, Academic Press, New York, 1973.

[8] C. S. Kubrusly and O. L. V. Costa, "Mean Square Stability Conditions for Discrete Stochastic Bilinear Systems," *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 11, pp. 1082-1087, November 1985.

[9] J. Liu, "On the Existence of a General Multiple Bilinear Time Series," *Journal of Time Series Analysis*, Vol. 10, No. 4, pp. 341-355, 1989.

Table I
Coefficients of the Unknown Bilinear System
Used in the Experiments

$a_0 = 1.0$	$a_1 = 1.0$	$a_2 = 1.0$	$b_1 = 0.5$	$b_2 = -0.5$	
$c_{0,1} = 0.3$	$c_{0,2} = 0.1$	$c_{1,1} = -0.2$	$c_{1,2} = -0.2$	$c_{2,1} = 0.1$	$c_{2,2} = 0.3$

Table IV
Mean-Squared Error

	ABF1		ABF2		ABF3		ABF4	
	ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1
White Noise	1.029×10^{-3}	0.1050	1.029×10^{-3}	0.1051	1.014×10^{-3}	0.1036	1.032×10^{-3}	0.1054
Colored Noise	1.056×10^{-3}	0.1031	1.056×10^{-3}	0.1030	1.059×10^{-3}	0.1034	1.057×10^{-3}	0.1109

Table II
Mean and Variance of Filter Coefficients
(Measurement Noise is White)

		ABF1		ABF2		ABF3		ABF4	
		ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1
a_1	Mean	0.9999	1.0057	1.0001	1.0089	0.9994	1.0077	0.9999	1.0074
	Variance	4.78×10^{-6}	3.80×10^{-4}	5.12×10^{-6}	4.64×10^{-4}	4.85×10^{-6}	4.77×10^{-4}	4.73×10^{-6}	4.77×10^{-4}
b_1	Mean	0.5004	0.4943	0.5003	0.4914	0.5001	0.4957	0.5016	0.4923
	Variance	4.79×10^{-6}	5.93×10^{-4}	5.02×10^{-6}	5.97×10^{-4}	3.23×10^{-6}	3.98×10^{-4}	4.86×10^{-6}	6.48×10^{-4}
$c_{2,2}$	Mean	0.3001	0.2941	0.2996	0.2927	0.2999	0.2949	0.3015	0.2956
	Variance	5.15×10^{-6}	5.77×10^{-4}	5.74×10^{-6}	7.63×10^{-4}	6.57×10^{-6}	4.80×10^{-4}	5.22×10^{-6}	9.18×10^{-4}

Table III
Mean and Variance of Filter Coefficients
(Measurement Noise is Colored)

		ABF1		ABF2		ABF3		ABF4	
		ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1	ξ_o=0.001	ξ_o=0.1
a_1	Mean	1.0002	1.0027	1.0002	1.0047	0.9995	1.0044	1.0011	1.1000
	Variance	1.78×10^{-6}	2.09×10^{-4}	2.23×10^{-6}	2.28×10^{-4}	4.53×10^{-6}	5.31×10^{-4}	1.86×10^{-6}	5.07×10^{-4}
b_1	Mean	0.5006	0.4989	0.5006	0.4975	0.5005	0.4988	0.5010	0.4027
	Variance	1.59×10^{-6}	1.53×10^{-4}	1.89×10^{-6}	1.84×10^{-4}	2.43×10^{-6}	2.12×10^{-4}	1.65×10^{-6}	9.74×10^{-4}
$c_{2,2}$	Mean	0.3007	0.3018	0.3004	0.3006	0.3004	0.3032	0.3014	0.2220
	Variance	2.56×10^{-6}	2.33×10^{-4}	3.43×10^{-6}	2.85×10^{-4}	7.67×10^{-6}	5.50×10^{-4}	2.58×10^{-6}	6.09×10^{-4}