Nonlinear IIR Adaptive Filtering Using a Bilinear Structure

X.Y. Gao, W.M. Snelgrove, and D.A. Johns

University of Toronto, Toronto, Canada M5S 1A4

ABSTRACT

A new nonlinear IIR adaptive filter is presented in this paper. The algorithm is based on a Volterra polynomial realized by a set of bilinear systems. Each bilinear system corresponds to a cascade connection of linear systems and multipliers. This allows the extension of some linear adaptive concepts. Nonlinear IIR adaptive filters have an advantage over FIR Volterra filters in that in some applications they may result in less computation. The results of a simulation example show the applicability and practical convergence rate of the nonlinear IIR adaptive filters presented.

1. INTRODUCTION

Adaptive filters are increasingly important components of communications systems. Currently the majority are linear finite impulse-response (FIR) filters whose coefficients are updated by the least mean-square (LMS) algorithm [1].

Recent work has extended the LMS algorithm to the adaptation of recursive linear filters with infinite impulse responses (IIR filters). These filters allow reduced order in many practical applications, especially where the physical system to be modeled or corrected has an infinite impulse response with a long time constant.

Another line of research has extended linear FIR adaptive filters to the nonlinear case [3,4,6-8]. These systems are particularly applicable in the "weakly nonlinear" case where small amounts of low-degree distortion products need to be dealt with. Mathematically, these systems are generally studied using the Volterra series, which is an extension of Taylor series to cover systems with memory (internal states).

This paper combines these two lines of research, by presenting a nonlinear IIR adaptive filter. This encompasses the earlier approaches, and is applicable where linearization and short-response approximations start to fail.

The adaptive filter presented here uses a steepest-descent algorithm. The computational overhead is low, with the complexity of the gradient computation comparable to that of the filter itself. Furthermore, the filter's recursive structure itself reduces computation costs from those for long FIR transversal filters.

This paper derives general gradient equations and also derives systems to implement these equations. The algorithm is then demonstrated on a simple model-matching problem with a second-degree nonlinearity.

2. THE VOLTERRA SERIES AND ITS BILINEAR REPRESENTATION

For a weakly nonlinear system, the output $y(k)$ can be expanded into a Volterra series in terms of the input signal $u(k)$

$$
\begin{aligned}
y(k) &= \sum_{n=0}^{\infty} y_n(k) \\
&= h_0 + \sum_{i_1=0}^{\infty} h_1(i_1)u(k-i_1) + \sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty} h_2(i_1,i_2)u(k-i_1)u(k-i_2) + \cdots \\
&\quad + \sum_{i_1=0}^{\infty}\cdots\sum_{i_n=0}^{\infty} h_n(i_1,i_2,\cdots,i_n)u(k-i_1)u(k-i_2)\cdots u(k-i_n) + \cdots \qquad (1)
\end{aligned}
$$

where $h_0, h_1, \ldots, h_n, \ldots$, are called the kernels of the system, and y_n indicates the nth order term which has the kernel h_n. In this paper, without loss of generality, the constant zeroth-order term h_0 will be taken to be zero. The first-order term, y_1, is the convolution for a linear system which has the impulse response h_1. The second-order term, y_2, together with the higher order terms y_n, models the nonlinearity. If the h_2 and higher order kernels are zero, the system is reduced to a linear system.

The system corresponding to the nth order term in (1) is called a degree-n homogeneous system since application of the input $\alpha u(k)$, where α is a scalar, yields the output $\alpha^n y_n(k)$, where $y_n(k)$ is the response to $u(k)$. Equation (1) shows that by means of the Volterra series, a nonlinear system is represented by a sum of homogeneous systems with consecutive degrees.

Under the assumption that the regular transfer function is recognizable [5], a degree-n homogeneous system can be represented by a bilinear system

$$
\begin{aligned}
\mathbf{x}_1(k+1) &= \mathbf{A}_1\mathbf{x}_1(k) + \mathbf{B}_1 u(k) \\
\mathbf{x}_2(k+1) &= \mathbf{A}_2\mathbf{x}_2(k) + \mathbf{B}_2\mathbf{x}_1(k)u(k) \\
&\ \ \vdots \\
\mathbf{x}_n(k+1) &= \mathbf{A}_n(k)\mathbf{x}_n(k) + \mathbf{B}_n\mathbf{x}_{n-1}(k)u(k) \\
y_n(k) &= \mathbf{C}_n\mathbf{x}_n(k) \qquad (2)
\end{aligned}
$$

This bilinear system corresponds to a cascade connection of multi-input , multi-output linear systems and vector multipliers as shown in Fig. 1.

In practice, a Volterra series can be truncated into a Volterra polynomial with the first few most significant terms. Most of the Volterra filters previously presented were developed based on the quadratic Volterra polynomial [3, 4, 6, 8].

3. ADAPTIVE VOLTERRA FILTERS

The adaptive Volterra filters presented here have a structure as shown in Fig.2. The filtering system consists of a set of homogeneous systems. The filtering system is adapted using the state signals and the error signal which is the difference between the desired signal, $\delta(k)$, and the filtering system output, $y(k)$.

ISCAS '89

In order to adapt the system coefficients, we need to know the gradient of the filtering system output with respect to the coefficients. Assume the system is represented by an N-term Volterra polynomial, and p is a coefficient of the degree-n homogeneous system, then the gradient of the system output with respect to the coefficient p can be written as

$$\frac{\partial y}{\partial p} = \frac{\partial}{\partial p}\sum_{i=1}^{N} y_n$$

$$= \frac{\partial y_n}{\partial p} \qquad (3)$$

since the coefficient p is not related to the outputs of other homogeneous systems. Thus the gradient of the overall filtering system output with respect to the coefficient p of the degree-n homogeneous system is equal to the gradient of the homogeneous system output with respect to p.

An efficient algorithm for gradient evaluation of the linear system has been presented in [2]. We will employ that approach to compute the gradients for our degree-1 homogeneous system. The details of this approach will not be repeated here. We will concentrate on gradient evaluation for higher degree homogeneous systems.

It is not necessary to adapt the $\mathbf{B}_1$ vector in (2) since we can always choose $\mathbf{B}_1$ as a constant vector, for example, $\mathbf{B}_1 = (1\,0\,0 \cdots 0)^T$ [5], thus we do not need to compute the gradients for the elements of $\mathbf{B}_1$.

In terms of state vectors, the solution of the degree-n ($n \geq 2$) homogeneous system described by the bilinear equation in (2) can be written as

$$y_n(k) = \mathbf{C}_n \mathbf{x}_n(k) \qquad (4)$$

for $m = n$, and

$$y_n(k) = \mathbf{C}_n \Phi_n(k) \otimes \mathbf{B}_n u(k) [\Phi_{n-1}(k) \otimes \mathbf{B}_{n-1} u(k)$$
$$(\cdots \Phi_{m+1}(k) \otimes \mathbf{B}_{m+1} u(k) \mathbf{x}_m(k) \cdots)]$$
$$m = 1, \cdots, n-1 \qquad (5)$$

where $\Phi_m(k) = \mathbf{A}_m^{k-1}$ is the transition matrix of the *mth* linear system and $\otimes$ indicates convolution.

From (4), it is clear that

$$\frac{\partial y_n(k)}{\partial c_{nj}} = x_{nj} \qquad (6)$$

where c_{nj} and x_{nj} indicate the *jth* elements of $\mathbf{C}_n$ and $\mathbf{x}_n$, respectively.

Suppose p is an element of the matrix $\mathbf{A}_m (m \geq 1)$ or $\mathbf{B}_m (m \geq 2)$ of the *mth* linear system of the degree-n homogeneous system in (2). Then, from (4), we have

$$\frac{\partial y_n(k)}{\partial p} = \mathbf{C}_n \frac{\partial \mathbf{x}_n(k)}{\partial p} \qquad (7)$$

for $m = n$, and from (5) we have

$$\frac{\partial y_n(k)}{\partial p} = \mathbf{C}_n \Phi_n(k) \otimes \mathbf{B}_n u(k) [\Phi_{n-1}(k) \otimes \mathbf{B}_{n-1} u(k)$$
$$(\cdots \Phi_{m+1}(k) \otimes \mathbf{B}_{m+1} u(k) \frac{\partial \mathbf{x}_m(k)}{\partial p} \cdots)]$$
$$m = 1, \cdots, n-1 \qquad (8)$$

since the coefficient p of the *mth* linear system is not related to the signal $u(k)$, nor to the matrices $\mathbf{C}_n$, Φ_i and $\mathbf{B}_i$ $(i > m)$. Thus we need to evaluate the gradients of the internal state vectors $\partial \mathbf{x}_m(k)/\partial p$. Let

$$\mathbf{P}_{mij}(k) = \frac{\partial \mathbf{x}_m(k)}{\partial a_{mij}} \quad m = 1,2, \cdots, n, \; i = 1,2, \cdots, n_m, \; j = 1,2, \cdots, n_m$$

$$\mathbf{R}_{mij}(k) = \frac{\partial \mathbf{x}_m(k)}{\partial b_{mij}} \quad m = 2,3, \cdots, n, \; i = 1,2, \cdots, n_m, \; j = 1,2, \cdots, n_{m-1}$$

where a_{mij} and b_{mij} are the elements on the *ith* row and *jth* column of the matrices $\mathbf{A}_m$ and $\mathbf{B}_m$, respectively. It can be shown, by differentiating both sides of the equation for the *mth* linear system in (2) and rearranging the terms, that $\mathbf{P}_{mij}$ and $\mathbf{R}_{mij}$ satisfy the following two equations, respectively

$$\mathbf{P}_{mij}(k+1) = \mathbf{A}_m \mathbf{P}_{mij}(k) + \mathbf{e}_i x_{mj}(k), \quad m = 1,2, \cdots, n,$$
$$i = 1,2, \cdots, n_m, \; j = 1,2, \cdots, n_m \qquad (9)$$

$$\mathbf{R}_{mij}(k+1) = \mathbf{A}_m \mathbf{R}_{mij}(k) + \mathbf{e}_i x_{(m-1)j}(k) u(k), \quad m = 2,3, \cdots, n,$$
$$i = 1,2, \cdots, n_m, \; j = 1,2, \cdots, n_{m-1} \qquad (10)$$

where $\mathbf{e}_i$ is a vector with the *ith* element equal to unity and others zero, $x_{(m-1)j}(k)$ and $x_{mj}(k)$ are the *jth* elements of the vectors $\mathbf{x}_{m-1}(k)$ and $\mathbf{x}_m(k)$, respectively. These equations indicate that the gradient signals can be obtained by creating new systems.

Using the steepest descent method, a coefficient p of the adaptive filter is updated in the following way:

$$p(k+1) = p(k) - \mu \frac{\partial E[e^2(k)]}{\partial p(k)} \qquad (11)$$

where $e(k)$ is the error signal which is the difference between the reference signal, $\delta(k)$, and the filtering system output, $y(k)$, E denotes the mathematical expectation, and μ is a step size to control convergence of the algorithm. With the LMS algorithm, the mean value of the squared error is approximated by the instantaneous value of the squared error. With this approximation and the fact that the reference signal $\delta(k)$ is not a function of $p(k)$, (11) can be written as

$$p(k+1) = p(k) + 2\mu e(k) \frac{\partial y(k)}{\partial p(k)} \qquad (12)$$

If every element of an $n \times n$ matrix A is to be adapted, there are n^2 elements to be adapted, and equation (9) shows that evaluating the gradient of the state vector with respect to an element of the corresponding A matrix involves convolution which is computationally expensive. Thus the computational cost for adapting all the elements of the A matrix is great, particularly when the matrix has a high dimension. We will show that it is not necessary to adapt all the elements of an A matrix.

Consider a multi-input and multi-output linear system

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x} \qquad (13)$$

where the vectors $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{y}$ have n, n_u, and n_y dimensions, respectively, and the matrices A, B, and C have corresponding dimensions. This is a general description of a linear system. To represent the *mth* linear system in (2), we can let the matrix $\mathbf{C}_m$ be an identity matrix so that the output vector is the state vector. The poles of the system transfer function of (13) are determined by the matrix A. It is well known that an arbitrarily given set of the poles can be realized by either of the two special forms of the A matrix:

$$\begin{bmatrix} 0 & 0 & . & 0 & a_1 \\ 1 & 0 & . & 0 & a_2 \\ 0 & 1 & . & 0 & a_3 \\ . & . & . & . & . \\ 0 & 0 & . & 1 & a_n \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 & . & 0 \\ 0 & 0 & 1 & . & 0 \\ 0 & 0 & 0 & . & 0 \\ . & . & . & . & . \\ a_1 & a_2 & a_3 & . & a_n \end{bmatrix}$$

(217) 333-4789 Further information can be obtained from Dr. W. Kenneth Jenkins.

With either of these forms of the **A** matrix, only n elements are required to adapt. The adaptation will be called single column or single row adaptation, if the first or the second form of the **A** matrix is used.

For the first linear system of the homogeneous system in (2), we need only to evaluate the gradients with respect to the n_1 elements of the $\mathbf{A}_1$ matrix with one of the special forms since $\mathbf{B}_1$ is fixed. Generally, the last linear system is a multi-input system with n_{n-1} inputs and has at most $n_n \times n_{n-1}$ zeros. The number of elements of $\mathbf{B}_n$ is $n_n \times n_{n-1}$, which is equal to the required number of degrees of freedom for adapting the zeros. However, both $\mathbf{C}_n$ and $\mathbf{B}_n$ contribute to the adaptation of the zeros, thus there is redundancy in terms of degrees of freedom. Therefore only $\mathbf{B}_n$ has to be adapted. In the special case when $n_{n-1} = 1$, i.e. the last linear system has a single input, it is preferred to adapt $\mathbf{C}_n$ rather than $\mathbf{B}_n$ since evaluation of gradients for the elements of $\mathbf{C}_n$ does not involve convolution. We also need to get the gradients for the n_n elements of $\mathbf{A}_n$. For the rest of the linear systems, the gradients for the n_m elements of the $\mathbf{A}_m$ and all the elements of the $\mathbf{B}_m$ matrix are required.

The implementations of the gradient evaluation for the elements of $\mathbf{A}_m$ and $\mathbf{B}_m$ are shown in Fig. 3a and Fig. 3b, respectively. Fig. 3 is drawn according to (9) and (10). The boxes labeled " $\mathbf{A}_m$, **I** " in Fig. 3b represent multi-input linear systems with the system feedback matrix $\mathbf{A}_m$ and system input matrix **I** which is the identity matrix with a dimension of $n_m \times n_m$. The output matrix $\mathbf{R}_{mj}$ $(j = 1, 2 \cdots n_{m-1})$ of the linear systems are defined as $\mathbf{R}_{mj} = (\mathbf{R}_{m1j}\ \mathbf{P}_{m2j}\ \cdots\ \mathbf{R}_{mn_mj})$. The gradient evaluation for the elements of $\mathbf{C}_n$ is straightforward.

4. SIMULATION EXAMPLE

The adaptive algorithm for nonlinear IIR filters was used in a system identification problem as shown in Fig.4. The system to be identified was described by a Volterra polynomial with two terms: a degree-1 homogeneous system, i.e. a linear system, and a degree-2 homogeneous system. The linear system had the following matrices:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.383092 & -0.813834 & 0.640753 \end{bmatrix}$$

$$\mathbf{B}_1 = (0\ \ 0\ \ 1)^T$$

$$\mathbf{C}_1 = (0.163072\ \ 0.181138\ \ 0.352637)$$

The degree-2 system had the following matrices:

$$\mathbf{A}_1 = [-0.3] \quad \mathbf{B}_1 = [1]$$

$$\mathbf{A}_2 = [0.2] \quad \mathbf{B}_2 = [1.5]$$

$$\mathbf{C}_2 = [0.1].$$

For the linear system, $\mathbf{C}_1$ was adapted, and single row adaptation was performed for the $\mathbf{A}_1$ matrix, while $\mathbf{B}_1$ was kept fixed. For the degree-2 homogeneous system, $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{C}_2$ were adapted, while $\mathbf{B}_1$ and $\mathbf{B}_2$ were fixed. The initial values of all the elements to be adapted were zero. The step size parameters (μ) for the linear system and the degree-2 homogeneous system were chosen as 0.01 and 0.005. The filter system converged to the proper parameters with four digits of accuracy for about $30k$ iterations.

As discussed before, we have used the instantaneous squared error, $e^2(k)$, to approximate the mean square error, $E(e^2(k))$. Therefore, the exact gradient of the squared error we use should approximate the gradient of the mean square error. To show this, we let both the reference and adaptive systems have only the degree-2 homogeneous systems, and only $\mathbf{A}_1$ and $\mathbf{A}_2$ be adapted. The mean square error contour superimposed with the adaptation path taken is plotted in Fig.5. It is seen that the adaptation path is approximately normal to the mean square error contours. This shows that the search direction is close to the steepest descent direction, and the gradient of the squared error is a close approximation of the gradient of the mean error square.

5. CONCLUSIONS

A Volterra polynomial, that is, truncated Volterra series, gives a good description of weakly nonlinear systems. On the basis of the Volterra polynomial, which can be realized by a set of bilinear systems, an adaptive algorithm has been developed for nonlinear IIR filters, along with low-cost gradient evaluation formulas. The Volterra filters are composed of a set of bilinear systems corresponding to a cascade connection of linear filters and multipliers. This is a pleasing feature since it enables us to extend some concepts of linear adaptive IIR filters. The nonlinear IIR filters with bilinear structure have advantages over the Volterra FIR filters since the IIR filters do not assume that the system has a finite memory, and may therefore require less computation. The results of the simulation example show the applicability and practical convergence rate of the Volterra IIR filters.

References

[1] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, New Jersey: Prentice-Hall, 1985.

[2] D.A. Johns, W.M. Snelgrove, and A.S. Sedra, "State-Space Adaptive Recursive Filters", *1988 IEEE ISCAS* ,1988, pp.2153-2156.

[3] T. Koh, and E.J. Powers, "Second-Order Volterra Filtering and Its Application to Nonlinear System Identification", *IEEE Trans. on Acoustics, Speech and Signal Processing* , vol. ASSP-33, Dec. 1985, pp. 1445 - 1455.

[4] C.E. Davila, A.J. Welch, and H.G. Rylander, " A Second-Order Adaptive Volterra Filter with Rapid Convergence", *IEEE Trans. on Acoustics, Speech and Signal Processing* , vol. ASSP-35, Sept. 1987, pp.1259-1263.

[5] W.J. Rugh, *Nonlinear system Theory — the Volterra/Wiener Approach* , Baltimore: the Johns Hopkins University Press, 1981.

[6] G.L. Sicuranza and G. Ramponi, "Adaptive Nonlinear Digital Filters Using Distributed Arithmetic", *IEEE Trans. Acoustics, Speech, and Signal Processing* , vol.ASSP-34, June 1986, pp.518-526.

[7] M.G. Bellanger, *Adaptive Digital Filters and Signal Analysis*, New York: Marcel Dekker, Inc., 1987.

[8] D. Mansour and A.H. Gray, "Frequency Domain Non-linear Adaptive Filter", *1981 IEEE ICASSP* , 1981, pp.550-553.

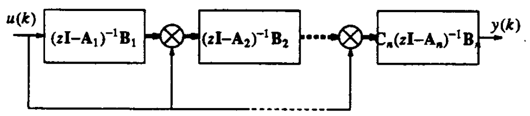

Fig.1 A degree-n homogeneous system represented by a bilinear system.

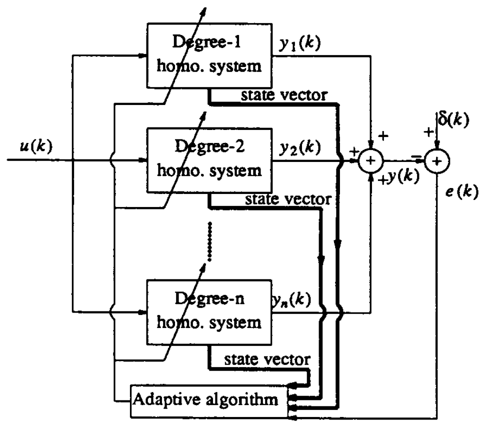

Fig.2 The general structure of an adaptive Volterra filter.

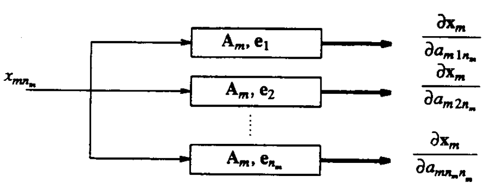

Fig. 3 a Gradient evaluation for the elements of the last column of A_m.

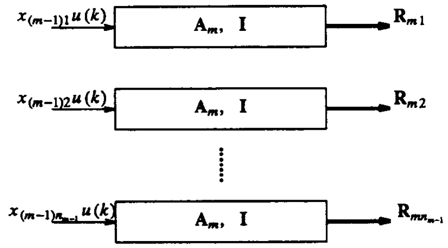

Fig. 3 b Gradient evaluation for the elements of B_m

Fig. 3 Implementation of gradient evaluation for the *mth* linear system of the degree-n homogeneous system.

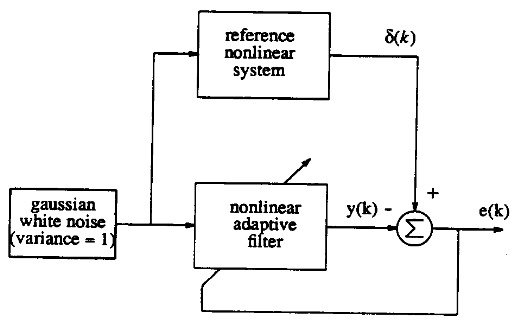

Fig.4 System identification application used for simulations

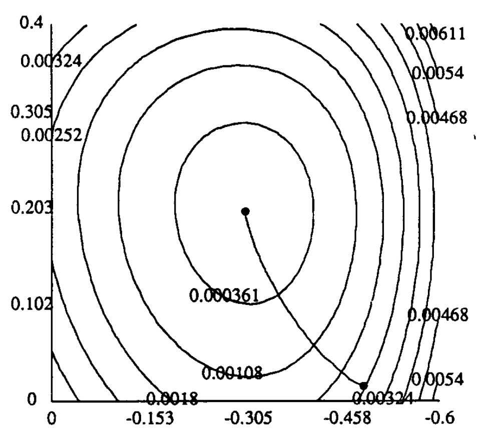

Fig 5 Error contour and the adaptation path.

(217) 333-4789 Further information can be obtained from Dr. W. Kenneth Jenkins.