Transactions Briefs

Adaptive Nonlinear Recursive State-Space Filters

Frank X. Y. Gao and W. Martin Snelgrove

Abstract-Adaptive **nonlinear filters previously reported often employ truncated Volterra series and have a finite impulse response (FIR). This paper introduces a nonlinear state-space structure for adaptive nonlinear filters. The adaptive filters are recursive and thus generally have an infinite impulse response (IIR). They are expected to be useful for many applications and are especially attractive for those with long memories where adaptive nonlinear FIR filters are too expensive to use. Efficient methods, which significantly reduce computation for gradients, have been developed to facilitate further their application in real-time signal processing. Numerical simulations have been performed to demonstrate the properties of the proposed algorithms.**

I. INTRODUCTION

A significant amount of research has been reported on adaptive nonlinear filters [1]-[16]. Those previously reported are often, directly or indirectly, based on Volterra theory and have finite impulse responses. They can be considered as extensions of adaptive linear FIR transversal filters to nonlinear problems. Adaptive nonlinear FIR filters share advantages and disadvantages with their linear counterparts. The problem of computation cost in the case of adaptive nonlinear FIR filters is much more serious than that in the case of adaptive linear FIR filters since their cost increases superlinearly, rather than linearly, with system memory length.

The computational disadvantage of adaptive nonlinear FIR filters can be easily shown by a simple example. Consider a nonlinear first-order physical system, with quadratic nonlinearity:

$$x_p(k+1) = a_p x_p(k) + b_p u(k) + p_{p1} x_p^2(k) + p_{p2} u(k) x_p(k)$$

$$y_p(k) = c_p x_p(k) \tag{1}$$

where x_p and y_p[1] are the state variable and the output variable, respectively.

The parameters used in the simulations were $a_p = -0.9$, $b_p = 0.8$, $p_{p1} = 0.01$, $p_{p2} = 0.03$, and $c_p = 1$. The ratio of the mean square of nonlinear component to that of the linear component of the system output, $E\left((0.01x_p^2 + 0.03ux_p)^2\right)/E\left((-0.9x_p + 0.8u)^2\right)$, is $-21dB$ for a Gaussian white input with a unit variance. An adaptive Volterra FIR filter was used to identify the system described in (1). Three tests were performed with different filter orders and the results are presented in Table I. For more details of the tests, the reader is referred to [16].

In Test 3, for example, the adaptive filter performed 7.84k multiplications per sample, which is computationally demanding. It is challenging to implement such an adaptive filter on a digital signal processor. For example, a Motorola DSP56001 chip can perform 10.25 million multiplications per second. It is able to perform 232 multiplications per sample at the 44.1 kHz compact-disc rate, or 1281 multiplications per sample at the 8 kHz speech rate. A single chip clearly cannot handle the computation load required by the adaptive nonlinear FIR filter in this example (tests 2 and 3) even at the speech rate. Convergence was also slow and final MSE was only fair in this example.

It is well known that adaptive linear IIR filters have a potential computational advantage over adaptive linear FIR filters. Similarly, we can expect that an adaptive nonlinear IIR filter is potentially more economical than an adaptive nonlinear FIR filter. Very few results have been reported on adaptive nonlinear IIR filters in the context of signal processing. An adaptive nonlinear IIR filter was presented in [14] using the Volterra series with a bilinear structure. Adaptive nonlinear recursive state-space (ANRSS) filters, first introduced in [15] [16], are more general in form than the adaptive nonlinear IIR filter in [14]. To make comparison between an adaptive nonlinear FIR filter and an ANRSS filter, numerical tests, corresponding to those in Table I, have been performed and the results will be presented in Section IV. These results indicate that for the first-order example, an ANRSS filter is able to match the reference physical system perfectly, with 0.4% of the computation required by the adaptive nonlinear FIR filter per iteration and with 6% of its convergence time.

One potential application of ANRSS filters is echo cancellation in telecommunications systems. The most notable sources of nonlinearity include the D/A converter [8], [11] and the line driver [11] at the transmission end. Since the linear part of the channel is often better approximated by a pole-zero model [2], the echo channel may be modeled as a nonlinear memoryless system followed by a linear dispersive system described by an IIR transfer function. Another potential application is identification and linearization of a loudspeaker, which is discussed in detail in Section IV.

In exchange for greatly improved efficiency, ANRSS filters have (in common with IIR adaptive filters generally [19], [20]) more complex mathematical behavior: there may be local minima, and either the filter itself or the overall algorithm may go unstable. For nonlinear systems, there is the additional complication that the best efficiency is obtained only when the mathematical form of the nonlinearity is known in advance. We believe that this trade-off is favorable for ANRSS filters for real-time applications involving "fine-tuning" or tracking of the parameters of systems with well-understood physics. An example is real-time modeling of loudspeakers, where a low-order recursive model with a small number of weak nonlinearities is well accepted, but where the exact values of the physical parameters vary slightly between loudspeakers and over time and temperature.

II. FILTER FORMULATION AND GRADIENT COMPUTATION

Suppose a physical system is described by a nonlinear recursive state-space equation of order n_p:

$$\boldsymbol{x}_p(k+1) = \boldsymbol{A}_p \boldsymbol{x}_p(k) + \boldsymbol{B}_p u(k) + \boldsymbol{g}_p\left(\boldsymbol{p}_p, u(k), \boldsymbol{x}_p(k)\right) \tag{2.a}$$

$$y_p(k) = \boldsymbol{C}_p^T \boldsymbol{x}_p(k) + d_p u(k), \tag{2.b}$$

Manuscript received August 27, 1990; revised December 6, 1993. This work was supported in part by NSERC Canada and by Ontario's Information Technology Research Centre. This paper was recommended by Associate Editor David C. Farden.

F. X. Y. Gao is with Gao Research & Consulting Ltd., 85 Dundalk Drive, Scarborough, Ontario, Canada M 1 P 4V 1.

W. M. Snelgrove is with the Department of Electronics, Carleton University, Ottawa, Ontario, Canada KIS 5B6.

IEEE Log Number 9405319.

[1] In this paper, the variables of physical systems have a subscript p, while those of adaptive filters do not.

TABLE I
NUMERICAL RESULTS FOR THE FIRST-ORDER EXAMPLE USING FIR FILTERS

Test	n_1	n_2	n_3	MSE(dB)	Multiplies (x10^3)/iteration	Iterations (x10^3)
1	70	0	0	-16	0.14	20
2	70	50	0	-37.7	5.65	80
3	70	50	10	-38.2	7.84	80

n_1, n_2 and n_3 are the orders of the FIR filters for the linear, quadratic and cubic nonlinearities respectively. When the orders n_2 and n_3 are zero, the corresponding nonlinear terms are zero.

where $\boldsymbol{A}_p$ is the system feedback matrix, $\boldsymbol{B}_p$ is the system input coefficient vector, $\boldsymbol{x}_p$ is the state vector, $\boldsymbol{g}_p$ is a nonlinear function without linear terms, and $\boldsymbol{p}_p$ is a vector of coefficients for the nonlinearity. The order of the system and the form of the nonlinear function $\boldsymbol{g}_p$ are assumed to be known. The exact values of $\boldsymbol{A}_p, \boldsymbol{B}_p$, $\boldsymbol{C}_p$, and $\boldsymbol{p}_p$ are not known. The nonlinear function $\boldsymbol{g}_p$ is assumed to be a truncated multi-dimensional Taylor series without linear terms and its coefficients are the elements of $\boldsymbol{p}_p$. Thus, the function is differentiable with respect to both $\boldsymbol{x}_p$ and $\boldsymbol{p}_p$.

An ANRSS filter employs the structure of the physical system in (2) and adapts its coefficients ***A*, *B*, *C*,** d, and $\boldsymbol{p}$ to minimize the mean square (MS) of the difference between its output and a desired signal. The parameters can be updated in a way similar to that for the well-known LMS algorithm:

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k + 2\mu e(k)\frac{\partial y(k)}{\partial \boldsymbol{w}}, \tag{3}$$

where the vector $\boldsymbol{w}$ includes all the coefficients to be adapted, μ is the step size, and e is the error signal, the difference between the desired signal and the filter output.

The gradients of the adaptive filter output with respect to an element of C and the feedthrough coefficient d can be easily written as

$$\frac{\partial y(k)}{\partial c_i} = x_i(k), \frac{\partial y(k)}{\partial d} = u(k),$$

where x_i is the ith element of the state vector x. If the gradients of the state vector x with respect to the elements of ***A*, *B*,** and ***p*** ***are*** defined as

$$\boldsymbol{F}_{ij}(k) = \frac{\partial \boldsymbol{x}(k)}{\partial a_{ij}}, \boldsymbol{Q}_i(k) = \frac{\partial \boldsymbol{x}(k)}{\partial b_i}, \text{and } \boldsymbol{H}_i(k) = \frac{\partial \boldsymbol{x}(k)}{\partial p_i},$$

where a_{ij} is the element on the ith row and the jth column of the ***A*** matrix, b_i and p_i *are* the ***ith*** elements of B and ***p*,** respectively, it can be shown that the gradients of the adaptive filter output with respect to these filter coefficients can be written as

$$\frac{\partial y(k)}{\partial a_{ij}} = \boldsymbol{C}^T\boldsymbol{F}_{ij}(k), \frac{\partial y(k)}{\partial b_i} = \boldsymbol{C}^T\boldsymbol{Q}_i(k), \text{and } \frac{\partial y(k)}{\partial p_i} = \boldsymbol{C}^T\boldsymbol{H}_i(k), \tag{4}$$

where $\boldsymbol{F}_{ij}(k)$, $\boldsymbol{Q}_i(k)$ and $\boldsymbol{H}_i$ ***are*** computed recursively from the following equations:

$$\boldsymbol{F}_{ij}(k+1) = \boldsymbol{A}\boldsymbol{F}_{ij}(k) + \boldsymbol{e}_i x_j(k) + \left(\frac{\partial \boldsymbol{g}(\boldsymbol{p}, u(k), \boldsymbol{x}(k))}{\partial \boldsymbol{x}(k)}\right)\boldsymbol{F}_{ij}(k) \tag{5}$$

$$\boldsymbol{Q}_i(k+1) = \boldsymbol{A}\boldsymbol{Q}_i(k) + \boldsymbol{e}_i u(k) + \left(\frac{\partial \boldsymbol{g}(\boldsymbol{p}, u(k), \boldsymbol{x}(k))}{\partial \boldsymbol{x}(k)}\right)\boldsymbol{Q}_i(k) \tag{6}$$

$$\boldsymbol{H}_i(k+1) = \boldsymbol{A}\boldsymbol{H}_i(k) + \left(\frac{\partial \boldsymbol{g}(\boldsymbol{p}, u(k), \boldsymbol{x}(k))}{\partial \boldsymbol{x}(k)}\right)\boldsymbol{H}_i(k) + \frac{\partial \boldsymbol{g}(\boldsymbol{p}, u(k), \boldsymbol{x}(k))}{\partial p_i} \tag{7}$$

where $\boldsymbol{e}_i$ is a vector with unity in the ith element and zero in others. Comparing (4), (5), (6), and (7) with (2), it is seen that the gradients are computed with systems very similar in structure to the adaptive filter itself.

Stability is a challenging issue for adaptive nonlinear IIR filters. In the following, some qualitative discussions are given for an ANRSS filter. Small step sizes can be employed to reduce the chances of instability. During adaptation, if the adaptive filter were to enter an unstable region, its output would become large and so would the MSE. A gradient-based adaptation algorithm tends to force the system back to the stable region. Small step sizes can normally prevent the system from getting too deep into an unstable region. The chances of instability can also be reduced if the starting point is chosen to be close to the optimal point.

Even with infinitely slow adaptation, the stability of a recursive nonlinear filter may be difficult to guarantee. In the important case, though, where a physical system is being modeled it may be possible to derive constraints on parameters from physical passivity considerations. In modeling a loudspeaker, for example, the original system is inherently passive. More generally, local stability of an adaptive nonlinear IIR filter may be verified by testing the stability of a linear model obtained by truncating a Taylor expansion of $\boldsymbol{g}$ around its operating point [16], Monitoring this "stability" would be expensive in general, but is trivial for important special cases such as first- and second-order systems.

As for adaptive linear IIR filters, ANRSS filters may sometimes get trapped in local minima. A general solution of this problem cannot be readily obtained, but a good starting point can minimize the chance of getting into a local minimum.

III. REDUCTIONS IN GRADIENT COMPUTATION

From the above discussion, we know that one gradient filter with complexity similar to that of the adaptive filter itself is needed to adapt each element of ***A*, *B*,** or ***p*.** This demands a substantial amount of computation. Two methods of reducing the computation will be discussed in this section.

3.1 *Keeping the Input Coefficient Vector Fixed*

The computation can be reduced if adapting ***B*** can be avoided. There is a way to do so if it is known which terms of $\boldsymbol{B}_p$ ***are zero*** and which are not, and the differences between $\boldsymbol{B}_p$ and ***B*** just result in scaled states and coefficients. This idea is best explained with an example. Suppose the physical system concerned is a second-order system described by

$$x_{p1}(k+1) = a_{p11}x_{p1}(k) + a_{p12}x_{p2}(k) + b_{p1}u(k) + p_{p1}x_{p1}^3(k)x_{p2}^2(k) \tag{8}$$

$$x_{p2}(k+1) = a_{p21}x_{p1}(k) + a_{p22}x_{p2}(k) + b_{p2}u(k) + p_{p2}u(k)x_{p1}^2(k) \tag{9}$$

$$y_p = c_{p1}x_{p1}(k) + c_{p2}x_{p2}(k) + d_p u(k), \tag{10}$$

where all the coefficients are unknown. Let us first assume that both elements of the input coefficient vector $\boldsymbol{B}_p$ ***are*** nonzero. For given $\boldsymbol{B}_p$ and ***B*,** there exist two nonzero scalars α_1 and α_2, which relate $\boldsymbol{B}_p$ and ***B*:**

$$b_1 = \alpha_1 b_{p1}, b_2 = \alpha_2 b_{p2}. \tag{11}$$

Next, multiplying both sides of (9) by α_1 and both sides of (10) by α_2 and performing some simple algebraic manipulations, we arrive at

$$x_1(k+1) = a_{11}x_1(k) + a_{12}x_2(k) + b_1u(k) + p_1x_1^3(k)x_2^2(k) \tag{12}$$

$$x_2(k+1) = a_{21}x_1(k) + a_{22}x_2(k) + b_2u(k) + p_2u(k)x_1(k) \tag{13}$$

$$y = c_1x_1(k) + c_2x_2(k) + du(k), \tag{14}$$

where

$$x_i = \alpha_i x_{pi},\ c_i = \frac{c_{pi}}{\alpha_i},\ \text{for } i = 1 \text{ and } 2,\ a_{11} = a_{p11},\ a_{12} = a_{p12}\frac{\alpha_1}{\alpha_2} \tag{15}$$

$$a_{22} = a_{p22},\ a_{21} = a_{p21}\frac{\alpha_2}{\alpha_1},\ d = d_p,\ p_1 = \frac{p_{p1}}{\alpha_1^2\alpha_2^2},\ p_2 = p_{p2}\frac{\alpha_2}{\alpha_1^2}. \tag{16}$$

The new system described by (12), (13), and (14) is obtained by scaling the original system. This scaling maintains the structure of the original system. From (15) and (16), we know that the $\alpha' s$ must be nonzero. Hence, if some elements of B_p are zero (or nonzero), the corresponding elements of B must also be zero (or nonzero), as suggested by (11).

Therefore, to use an adaptive nonlinear filter to match a physical system described by (8), (9), and (10), we can set the input coefficient vector of the adaptive filter to be a constant vector with the same zero-nonzero pattern as that of the physical system, and the adaptive filter can just adapt the feedback matrix A, the output coefficient vector C, the feedthrough coefficient d and the nonlinear coefficients p to match the physical system, resulting in a scaled system model. The n gradient filters for the input coefficient vector B are then not needed. It can be shown that this is generally true for the case where the nonlinear function vector $g_p(p_p, u(k), x_p(k))$ is an n-dimensional Taylor series without linear terms. A direct-form equation is an example where the zero-nonzero pattern of the input coefficient vector is known.

In practice, B would be set to estimates from the physics of the nonlinear system. This can provide a good starting point and avoid some numerical difficulties arising from scaling. Further, adapting B and C simultaneously would result in difficulties in convergence due to redundant degrees of freedom.

3.2 *The Approximate Stochastic-Gradient Method*

The technique of gradient approximation has been widely and successfully applied in many practical optimization problems [18], This technique can be applied to the ANRSS filters to reduce computation. If the system is weakly nonlinear (the magnitude of the signal from the nonlinear part g_p of the physical system is much smaller than that from the linear part $A_p x_p(k) + B_p u(k)$), we can compute the approximate gradients for filter coefficients by neglecting the nonlinear part, thus considering the gradient filters as linear.

When neglecting the nonlinearity, the gradients for A and B of an ANRSS filter can be computed like those of an adaptive linear recursive state-space filter [17].

$$F_j(k+1) = A^T F_j(k) + C x_j(k) \tag{17}$$

and

$$Q(k+1) = A^T Q(k) + C u(k), \tag{18}$$

where the ith elements of $Q(k)$ and $F_j(k)$ are $\partial y(k)/\partial b_i$ and $\partial y(k)/\partial a_{ij}$. One gradient filter is able to generate gradients for all the elements of one column of matrix A, and one gradient filter for all the elements of B, resulting in a significant reduction in the computation. Evaluation of the gradient for B is also discussed here since it may sometimes be necessary to adapt B. The approximate gradient for p can be computed from

$$H_i(k+1) = A H_i(k) + \frac{\partial g(p, u(k), x(k))}{\partial p_i}. \tag{19}$$

Although the nonlinearity is ignored when approximating gradients, it is still used for computing the adaptive filter output. If exact gradients for all coefficients adapted are used, the method will be referred to as the stochastic-gradient method. On the other hand, if approximate gradients for all coefficients other than C are used, it will be referred to as the approximate stochastic-gradient method. As for C, the exact gradient is easily available and therefore always used. The approximate stochastic-gradient method is only suitable for the case of weak nonlinearity, say where the signal power in nonlinear terms is less than 10% of that in linear terms. If the nonlinearity is strong, gradient accuracy will be degraded and the full stochastic-gradient method should be used instead.

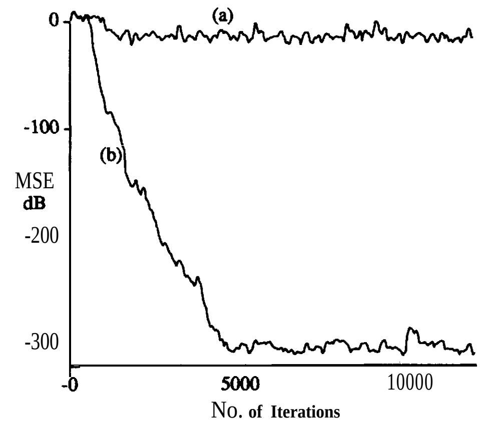

Fig. 1. Convergence curves for the first-order example. (a) Adapting linear coefficients only; (b) approximate stochastic-gradient method.

IV. NUMERICAL EXAMPLES

Two examples are shown to illustrate the utilization and performance of the adaptive filters proposed. The first example is the same as the one for adaptive FIR filters in Section I. The second example is identification and linearization of a loudspeaker model.

4.1 *Example l-First-Order System*

The input coefficient vector B_p of the first-order example in (1) has a known zero-nonzero pattern: the only element is always nonzero. Hence, the adaptive filter input coefficient b was fixed at unity and other coefficients were adapted. The physical system in (1) was used as the reference system. The adaptive filter updated its coefficients a, c, p_1, and p_2, with initial values being zero. The step sizes were $\mu_a = 0.0005$, $\mu_c = 0.01$, and $\mu_p = 0.0005$.

To show the effect of adapting linear coefficients only, a test was run. Curve (a) in NN was from this test. The MS error could go down to only about -15dB.

The approximate stochastic-gradient method was simulated next. The convergence curve is depicted in Fig. 1 as curve (b), which shows that the MSE was reduced from *OdB* to *-1OOdB* after lk iterations. Two contours have been drawn in Figs. 2 and 3 for the linear and nonlinear coefficients to show the performance surface and the adaptation behavior of the algorithm. Small step sizes were used so that the adaptation paths are smooth. It is obvious from the contour plots that the paths are generally normal to the contours, which is a characteristic of the steepest descent algorithms.

The stochastic-gradient method (without approximating gradients) was simulated next and very small differences between the results of the approximate stochastic-gradient method and the stochastic-gradient method were observed. The convergence curve and adaptation paths of the approximate stochastic-gradient method are slightly less smooth than those of the stochastic-gradient method since the

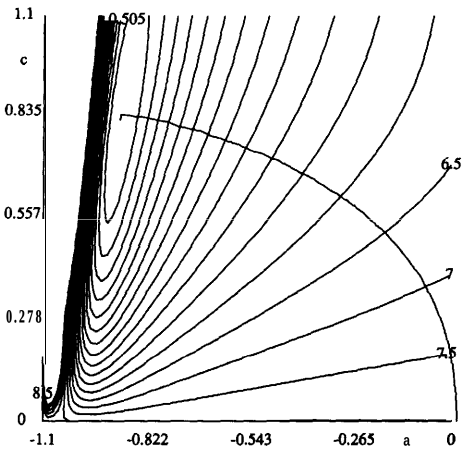

Fig. 2. Nonlinear state-space filter for the first-order example. $\mu_a = 7x10^{-5}, \mu_c = 7x10^{-5}$. Approximate stochastic-gradient method.

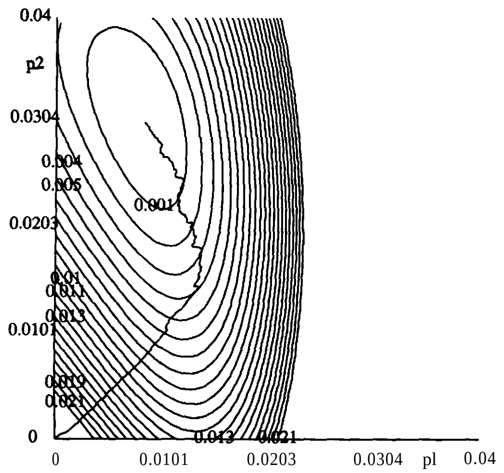

Fig. 3. Nonlinear state-space filter for the first-order example. $\mu_p = 5x10^{-6}$. Approximate stochastic-gradient method.

TABLE II
MAJOR RESULTS FOR THE FIRST-ORDER EXAMPLE USING THE ADAPTIVE NONLINEAR IIR AND FIR FILTERS

Test	MSE(dB)	Multiplication/iteration	Iterations
IIR	-300	33	5×10^3
FIR (case **3**)	-38.2	$7.84 x 10^3$	$80 x 10^3$

The number of multiplications per iteration given for the IIR filter is for the full stochastic-gradient method

nonlinearities neglected in computing gradients by the approximate stochastic-gradient method create noise in gradient computation.

We are now in a position to make a comparison between the results of the adaptive FIR filters and adaptive IIR filters on this first-order example. The major results for the adaptive nonlinear IIR and FIR filters are summarized in Table II. For this example, an ANRSS filter is able to match the reference physical system perfectly, with 0.4% of the computation required by the adaptive nonlinear FIR filter per iteration and with 6% of its convergence time. For this example, the adaptive IIR filter clearly outperformed the adaptive FIR filter.

4.2 *Example 2-Identification and Linearization of a Loudspeaker*

The nonlinear function $\boldsymbol{g}_p$ of the physical system in (2) brings in nonlinearity and causes distortion. The nonlinear term can be canceled

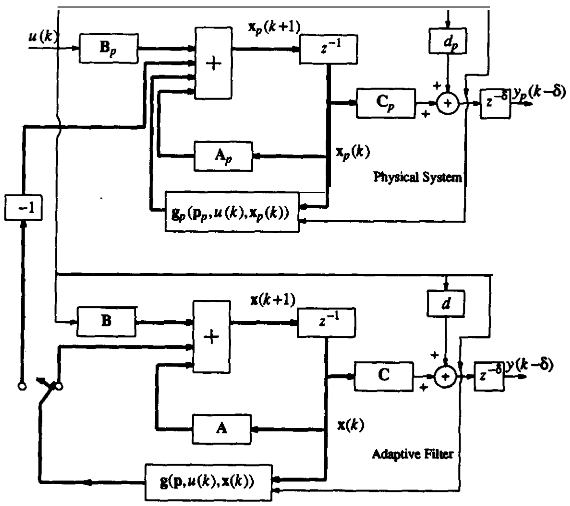

Fig. 4. The adaptive linearization scheme using the nonlinear state-space filter.

out by subtracting an estimate of it from the right-hand side of 2(a). The adaptive linearization scheme is illustrated in Fig. 4. The adaptive filter based on the model in (2) estimates the nonlinear coefficient vector $\boldsymbol{p}_p$ and the state vector $\boldsymbol{x}_p(k)$, which, together with $u(k)$, determine the estimate of $\boldsymbol{g}_p$.

The adaptive linearization process has two phases: first an identification phase, then a linearization phase. In the identification phase, the output of the nonlinear function $\boldsymbol{g}$ of the adaptive filter is fed to itself so that the adaptive filter is able to match the physical system. In the linearization phase, the switch of Fig. 4 will toggle so that $-\boldsymbol{g}(\boldsymbol{p}, u(k), \boldsymbol{x}(k))$ is fed to the physical system and thus linearizes the system. Note that switching out the nonlinearity makes the adaptive filter linear and enables its states to trace the linearized system. The term $z^{-\delta}$ in Fig. 4 models the possible delay between the system output and measured output.

The linearization scheme can be applied to a loudspeaker system. Assume that the nonlinearity of a loudspeaker is due to its suspension system. A loudspeaker can be modeled as [16]

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 \\ a_{p1} & ap2 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ b_{p2} \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ g(\mathbf{p}_p, \mathbf{x}_p(k)))^T \end{bmatrix}$$
$$y(k) = (10)^T \boldsymbol{x}(k) \tag{20}$$

where $g_p(\boldsymbol{p}_p, \boldsymbol{x}_p(k)) = p_{p1} x_{p1}^2(k) + p_{p2} x_{p1}^3(k)$ and $\boldsymbol{p}_p = (p_{p1} p_{p2})^T$. An estimate of the term $-\boldsymbol{g}_p(\boldsymbol{p}_p, \boldsymbol{x}_p(k))/b_{p2}$ is required to add to the input signal $u(k)$ to cancel out the nonlinearity of the original system.

Identification and linearization of a loudspeaker has been simulated. The parameters of the loudspeaker model were chosen as $a_{p1} = 0.3$, $a_{p2} = 0.2$, $p_{p1} = 0.006$, $p_{p2} = 0.03$, $b_{p2} = 0.6$, and $c_{p1} = 1$. That p_{p2} was chosen larger than p_{p1} was to be consistent with the fact that the cubic term is dominant in the suspension nonlinearity. The adaptive filter input coefficient vector was set to be a constant vector $(0\ 1)^T$. The adaptive filter updated its coefficients a_1, a_2, p_1, p_2, and c_1, with zero initial values. The step sizes were $\mu_a = 0.02$ for a_1 and $a_2, \mu_p = 0.001$ for p_1 and $p_2, \mu_c = 0.02$ for c_1. The delay in the air path was chosen as 50 sampling periods. In practice, this delay can be measured by feeding an impulse signal to the loudspeaker or using an adaptive linear transversal filter to estimate it. It is also possible to cascade an adaptive linear transversal filter with an ANRSS filter to perform on-line estimation of the delay. The interaction between the two cascaded filters may influence the convergence of the system, but we chose to leave that as an issue for future research.

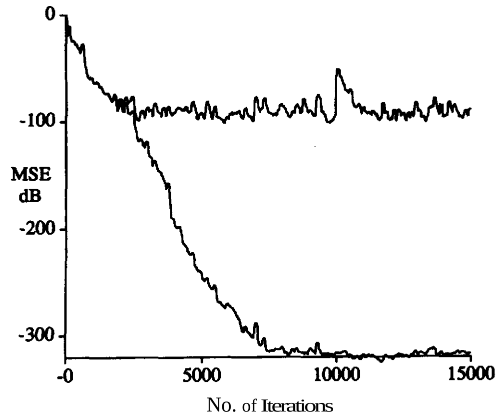

Fig. 5. Convergence curves for the loudspeaker example. The lower curve is for the case where the filter's model is the same as tbat of the system's. The upper curve for the case where the filter's model is not exactly the same as that of the system's. In both cases, the approximate stochastic-gradient method was used.

Both the stochastic-gradient method and the approximate stochastic- gradient method were run. The convergence curves of the two methods are similar, with differences of a few dB in the final stage of the runs. Only the curve for the approximate stochastic-gradient method is shown here in Fig. 5 for identification up to 30k iterations. The linearization took effect at 10k iterations. The nonlinear distortion was reduced from -3dB to -310dB This distortion reduction is so good that it can only be achieved in simulation, and some factors, such as measurement noise and model mismatch, will degrade the performance in a practical situation.

It is interesting to see the performance of the algorithms when the model is not exact. Suppose that a loudspeaker also has a nonzero quartic term in the nonlinear feedback term, that is $g_p(\boldsymbol{p}_p, \boldsymbol{x}_p(k)) = p_{p1}x_{p1}^2(k) + p_{p2}x_{p1}^3(k) + p_{p3}x_{p1}^4(k)$, but the adaptive filter just has a cubic feedback nonlinearity with $g(\boldsymbol{p}, \boldsymbol{x}(k)) = p_1 x_1^2(k) + p_2 x_1^3(k)$. For a numerical experiment, the parameter p_{p3} was chosen to be 2×10^{-5}. Other parameters were the same as before. The convergence curve is plotted in Fig. 5 for the approximate stochastic-gradient method. The adaptive filter worked well and reduced MSE to about -90dB a residual floor determined by the term in the loudspeaker which was not modeled by the adaptive filter. The nonlinear distortion was reduced from about -23dB to -49dB

V. SUMMARY

ANRSS filters have been introduced in this paper, which are computationally more attractive than adaptive nonlinear FIR filters for some applications. To take advantage of ANRSS filters, one has to have some knowledge of the system: most importantly its mathematical structure. Knowledge of the estimated values of the system parameters can be used to improve the filter performance. These requirements are practical since the physics of the system is normally understood and existing identification techniques can be used to verify models and to obtain initial parameter estimates.

Efficient adaptation algorithms have been developed for ANRSS filters. It has been shown that the input coefficient vector need not be adapted if we know the zero-nonzero pattern of the input coefficient vector of the physical system to be matched. The gradients of the adaptive filter coefficients can be efficiently computed by neglecting the nonlinearity in the system in the case of weak nonlinearity. Although the nonlinearity is neglected when computing gradients, it is still used to evaluate the adaptive filter output. The approximate stochastic-gradient method performed quite well in our simulations.

ACKNOWLEDGMENT

The author wishes to acknowledge Dr. D. Johns of the University of Toronto for his valuable discussions and the reviewers and CAS Associate Editor Dr. D. C. Farden for their insightful criticisms which significantly enhance the quality of the paper.

REFERENCES

[1] G.L. Sicuranza and G. Ramponi, "Adaptive nonlinear digital filters using distributed arithmetic," ***IEEE Trans. Acoust., Speech, Signal Process.,*** vol. ASSP-34, pp. 518-526, June 1986
[2] E. Biglieri, A. Gersho, R. D. Gitlin, and T. L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," ***IEEE J. Select. Areas Commun.,*** vol. SAC-2, pp. 765-777, Sept. 1984.
[3] M. J. Coker and D. N. Simkins "A nonlinear adaptive noise canceler," in ***Proc. 1980 IEEE Int. "Conf. Acoustics, Speech, Signal Process.,*** 1980, pp. 470-473,
[4] D. Mansour and A. H. Gray, "Frequency domain non-linear adaptive filter," in ***Proc. I980 IEEE Int. Conf. on Acoustics, Speech, Signal Process., 198 1,*** pp. 550-553.
[5] D. D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission systems," ***Bell System Tech. J., vol. 57,*** pp. 2589-2611, Sept. 1978.
[6] Y. Lou, C. L. Nikias, and A. N. Venetsanopoulos, "VLSI array processing structure of quadratic digital filters with LMS algorithm," in ***Proc. I987 Int. Conf Acoustics, Speech, Signal Process., 1987,*** pp, 13941397.
[7] M. J. Smith, C. F. N. Cowan, and P. F. Adams, "Nonlinear echo cancellers based on transposed distributed arithmetic," ***IEEE Trans. Circuits Syst., vol.35*** pp. 6-18, Jan. 1988.
[8] O. Agazzi, D. G. Messerschmitt, and D. A. Hodges, "Nonlinear echo cancellation of data signals," ***IEEE Trans. Commun.*** , vol. COM-30, pp. 242I-2433, Nov. 1982.
[9] G. L. Sicuranza, A. Bucconi, and P. Mini, "Adaptive echo cancellation with nonlinear digital filters," in ***Proc. I984 IEEE Int. Conf Acoustics, Speech, Signal Process. ' 1984, pp.3.10.1-3.10.4.***
[10] H. Khorramabadi ***et al.*** "An ANSI standard ISDN transceiver chip set," in ***Proc. 1989 IEEE Int. Solid-State Circ. Conf, 1989,*** pp. ***256-257.***
[11] ***K.*** Murano, S. Unagami, and F. Amano, "Echo cancellation and applications," ***IEEE Commun. Mug.,*** vol. ***28,*** Jan. 1990, pp. 49-55.
[12] T. Koh, and E. J. Powers, "Second-order volterra filtering and its application to nonlinear system identification," ***IEEE Trans. Acoustics, Speech, Signal Process.,*** vol. ASSP-33, pp. 1445-1455, Dec. 1985.
[13] C. E. Davila, A. J. Welch, and H. G. Rylander, "A second-order adaptive volterra filter with rapid convergence," ***IEEE Trans. Acousticas Speech, Signal Process.,*** vol. ASSP-35, pp.l259-1263, Sept. 1987.
[14] F. X. Y. Gao, W. M. Snelgrove, and D. A. Johns, "Nonlinear IIR adaptive filtering using a bilinear structure," in ***Proc. I989 IEEE Int. Symp. Cinxits Syst.,*** May 1988, pp. 1740-1743.
[15] F. X. Y. Gao and W. M. Snelgrove, "Adaptive nonlinear state-space filters," ***Proc. IEEE Int. Symp. CircuitsSyst.,*** May 1990, pp. 3 122-3 125.
[16] F. X. Y. Gao, ***Adaptive Linear and Nonlinear Filters,*** Ph.D dissertation, University of Toronto, 1991.
[17] D, A. Johns, W. M. Snelgrove, and A. S. Sedra, "Adaptive recursive state-space filters using a gradient based algorithm," ***IEEE Trans. Circuits Syst.,*** vol. ***37,*** June 1990, pp. 673-684.
[18] T. A. J. Nicholson, ***Optimization in Industry.*** Longman Group Limited, 1971
[19] D. A. Johns, "Analog and digital state-space adaptive IIR filters," Ph.D. dissertation, University of Toronto, 1989.
[20] C. R. Johnson, Jr., "Adaptive IIR filtering: Current results and open issues," ***IEEE Trans. Inform. Theory,vol. IT-30,*** pp. 237-250, March 1984.