Adaptive Recursive State-Space Filters Using a Gradient-Based Algorithm

DAVID A. JOHNS, MEMBER, IEEE, W. MARTIN SNELGROVE, MEMBER, IEEE, AND ADEL S. SEDRA, FELLOW, IEEE

Abstract —**Adaptive recursive filters are often implemented using direct-form realizations. However, direct-form realizations are known to have very poor sensitivity and roundoff noise properties, especially in the case of oversampled filters. To aid in the search for better adaptive filter structures, this paper presents a method to obtain the gradients required to adapt general state-space filters. Unfortunately, the number of computations for this general case is quite high. To reduce the number of computations, two new state-space adaptive filters are introduced. One application where these new structures are shown to be useful is in oversampled filtering where an estimate of the final pole locations is known and the adaptive filter is required only to "fine-tune" the transfer function. It is shown that for this type of application, the new adaptive structures can have much improved adaptation rates and roundoff noise performance than their corresponding direct-form realizations.**

I. Introduction

ADAPTIVE FILTERS have gained widespread acceptance as a tool for system designers. Typically, adaptive filters are implemented as transversal FIR filters and the simple LMS adaptation algorithm [1] is used to adjust the zeros of the filter. However, since only the zeros of an FIR filter are adapted, high-order filters are often required to achieve satisfactory performance. In many cases, the order can be reduced significantly through the use of an adaptive IIR filter where both the poles and zeros are adapted. For this reason, there has been considerable interest in adapting recursive filters, as is evident from the fact that numerous algorithms have been proposed for direct-form realizations [2]–[7] and lattice form [8], [9] as well as biquad form [10].

This paper proposes a gradient search technique for adapting the poles and zeros of a general state-space recursive filter. Being able to adapt arbitrary state-space filters gives the designer the freedom to explore the performance advantages of different structures [11]. Unfortunately, the computation requirements to adapt a general state-space gradient filter is quite high. However, it will be shown that the number of computations can be reduced by adapting any single column of the feedback matrix, and in special cases, a single row. The importance of these new structures is that in oversampled applications where final pole locations can be estimated, adaptive filters with much better adaptation rates and roundoff noise performance than those based on the direct-form structure can be obtained. In other words, these new structures will be useful in applications where an approximate shape of the final transfer-function is known and the adaptive algorithm is required only to "fine-tune" the filter. One example of this type of application is in fixed-channel data equalization, where tuning of the adaptive filter need account only for manufacturing tolerances. The advantage of this approach over adaptive recursive lattice filters is the reduced amount of computation required to obtain the gradients.

It should be pointed out that the adaptive algorithms presented in this paper are all based on the steepest descent adaptive algorithm (i.e., gradient-based). Thus, using a small enough step size (choosing this step size is a nontrivial task; see, for example, [12]), the systems will converge to a minimum in the error performance surface, assuming filter instability does not occur. However, there are problems associated with using gradient-based algorithms when adapting poles, such as ensuring the filter remains stable and the possibility of converging to a local minimum. Ensuring the filter remains stable can be accomplished by using filter structures having a simple stability check. On the possibility of converging to a local minimum, it has been proven that increasing the filter order will remove local minima when the number of free parameters in the numerator of the adaptive filter is equal to the order of the filter [13], [14]. On the other hand, excessive filter order can cause difficulties with parameter convergence and internal stability. Although filter instability and global convergence are important issues, the purpose of this paper is to concentrate on the effects of using different structures in gradient-based adaptive filter implementations. The results of this paper indicate that when gradient-based adaptive algorithms are used, there are some definite advantages in using filter structures other than direct form. It is possible, of course, that similar advantages might occur if global convergence algorithms are also applied to structures other than direct form.

Manuscript received September 28, 1988; revised April 25, 1989 and November 13, 1989. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant A7395 and Grant A4894. This paper was recommended by Associate Editor R. Ansari.

The authors are with the Department of Electrical Engineering, University of Toronto, Toronto, Ont., Canada.

IEEE Log Number 9034903.

0098-4094/90/0600-0673$01.00 ©1990 IEEE

This paper is divided as follows. Some background on state-space systems is presented in Section II. Also, recently presented sensitivity formulas [15] relating the change in the filter output to the system coefficients are modified to a form more useful to our present needs. These formulas are required in order to find the gradients so that the performance surface of an adaptive filter may be descended. Section III describes the adaptation algorithm for a general state-space filter that, unfortunately, is computationally intensive. Section IV presents a single-column adaptation algorithm that reduces the computations required. Column adaptation tests are developed to help one check whether a column of a particular design can be adapted. In addition, a single-row adaptation structure is presented. Finally, in this section, qualitative arguments are given indicating the reason that the direct-form structure adapts slowly for oversampled applications, and a method is discussed for choosing the initial system of the single-column or single-row adaptive filter. A noise performance comparison among different filter structures is presented in Section V to illustrate the noise advantage of single-column (or row) adaptation. Simulation results for a number of examples are given in Section VI illustrating the possible convergence speed advantage.

II. State-Space Systems

This section will present some sensitivity formulas for state-space systems [15] that will allow us to adapt the coefficients of state-space filters. These formulas require the definition of two sets of intermediate functions. Also in this section will be a transformation of a state-space system such that the two sets of intermediate functions are exchanged.

An Nth-order state-space digital filter can be described by the following equations:

$$x(n+1) = Ax(n) + bu(n)$$
$$y(n) = c^T x(n) + du(n) \tag{1}$$

where $x(n)$ is a vector of N states, $u(n)$ the input, $y(n)$ the output and A, b, c, and d the coefficients relating these variables. The matrix A is $N \times N$, the vectors b and c are $N \times 1$ and d is a scalar. Using z transforms, the transfer function from the filter input to the output is easily derived as

$$\frac{Y(z)}{U(z)} = c^T(zI - A)^{-1}b + d. \tag{2}$$

From this equation, we see that the poles of the system are determined by the A matrix (the poles are simply the eigenvalues of A), whereas the zeros of the system are related to all four of the state-space matrices.

We now define two sets of intermediate functions, $\{F_i(z)\}$ and $\{G_i(z)\}$, which were originally presented in [16]. The first set, which we write as the vector $F(z)$, consists of the transfer-functions from the filter input to the filter states.

$$F_i(z) \equiv \frac{X_i(z)}{U(z)}. \tag{3}$$

This definition leads to the formula

$$F(z) = (zI - A)^{-1}b. \tag{4}$$

The second vector of functions, $G(z)$, is defined as the set of transfer-functions from the input of each of the delay operators to the output. Thus if we inject a signal $\epsilon_i(z)$ at the input of the ith delay operator, then

$$G_i(z) \equiv \frac{Y(z)}{\epsilon_i(z)}. \tag{5}$$

Using this definition, we have

$$G^T(z) = c^T(zI - A)^{-1}. \tag{6}$$

To obtain gradient signals, we use the sensitivity formulas in [15] (which originally were obtained using a signal-flow-graph approach) to relate the derivatives of the output signal with respect to each of the system coefficients, to the intermediate functions.

$$\frac{\partial Y(z)}{\partial A_{ij}} = G_i(z)X_j(z) \tag{7}$$

$$\frac{\partial Y(z)}{\partial b_i} = G_i(z)U(z) \tag{8}$$

$$\frac{\partial Y(z)}{\partial c_i} = X_i(z) \tag{9}$$

$$\frac{\partial Y(z)}{\partial d} = U(z). \tag{10}$$

From the preceding equations, it is obvious that the gradient signals required to adapt the c vector elements are available as the output states, $x(n)$, whereas the gradient signal for the d scalar is the input signal, $u(n)$. However, to create the gradient signals required to adapt the elements of the A matrix and b vector, a new system is required, having the intermediate functions from the input to the states equal to $G(z)$ of the original system. This new system can be found by utilizing a simple transformation that exchanges the two sets of intermediate functions [16]. Specifically, given a system $[A, b, c, d]$ with intermediate functions $F(z)$ and $G(z)$, we can create a new system such that

$$F_{\text{new}}(z) = G(z) \quad \text{and} \quad G_{\text{new}}(z) = F(z) \tag{11}$$

by arranging that coefficients of the new system are related to those of the original system by

$$A_{\text{new}} = A^T \quad b_{\text{new}} = c \quad c_{\text{new}} = b \quad d_{\text{new}} = d. \tag{12}$$

This result can be verified easily using the formulas in (4) and (6).

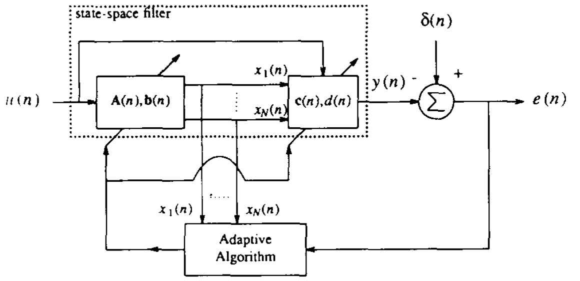


Fig. 1. Adaptive state-space filter. The filter is shown in two separate blocks corresponding to the state-space describing equations.

III. LMS Adaptive Algorithm for State-Space Filters

A block diagram of a state-space recursive adaptive filter is shown in Fig. 1, where the state-space coefficients now change with each time step and, hence, are functions of the time step "n." The state-space system is shown as two separate blocks that correspond with the state-space describing equations. Specifically, the feedback matrix, $\boldsymbol{A}$, and the input summing vector, $\boldsymbol{b}$, implement the first equation of a state-space system and create the state signals, $\boldsymbol{x}(n)$, as the outputs of the first block. These state signals together with the system input, u, are weighted using the output summing vector, $\boldsymbol{c}$, and the output scalar, d, to obtain the filter output, y, at the output of the second block. The error signal, $e(n)$, is the difference between the reference signal, $\delta(n)$, and the filter output, $y(n)$. During adaptation, coefficients of the adaptive filter are changed to minimize the mean-squared error signal, denoted as $E[e^2(n)]$. With the use of gradient signals, the steepest descent algorithm can be used to find a minimum of the mean-squared error performance surface. The algorithm for updating any coefficient, p, of the adaptive filter using the steepest descent algorithm is

$$p(n+1)=p(n)-\mu\frac{\partial E\left[e^2(n)\right]}{\partial p(n)} \tag{13}$$

where μ is a step-size parameter that controls convergence of the algorithm. With the LMS algorithm [1], the instantaneous value of the squared error signal is used to approximate the expected value. With such an approximation and using the fact that the error signal is the difference between the reference signal, $\delta(n)$ (which is not a function of p), and the output of the adaptive filter, $y(n)$, we have

$$p(n+1)=p(n)+2\mu e(n)\frac{\partial y(n)}{\partial p(n)}. \tag{14}$$

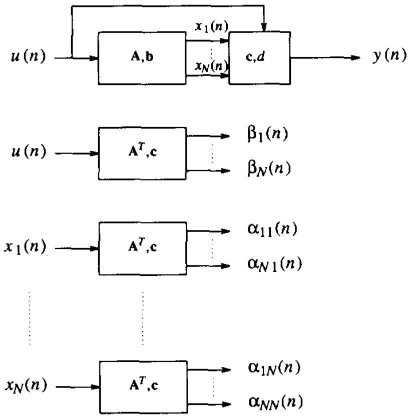


Fig. 2. Generating the gradients for a general state-space adaptive filter.

Now, as in previous publications [10], [12], we take the inverse z-transform of the gradient results found in the previous section[1] and substitute the resulting time-domain equivalent formulas in 14). This results in the following adaptation equations for the system coefficients:

$$A_{ij}(n+1)=A_{ij}(n)+2\mu e(n)\alpha_{ij}(n) \tag{15}$$

$$b_i(n+1)=b_i(n)+2\mu e(n)\beta_i(n) \tag{16}$$

$$c_i(n+1)=c_i(n)+2\mu e(n)x_i(n) \tag{17}$$

$$d(n+1)=d(n)+2\mu e(n)u(n) \tag{18}$$

where

$$\alpha_{ij}=g_i(n)\otimes x_j(n) \tag{19}$$

$$\beta_i=g_i(n)\otimes u(n) \tag{20}$$

and the symbol $\otimes$ denotes convolution.

Note that the adaptation equations for the elements of $\boldsymbol{A}$ and $\boldsymbol{b}$ involve convolution, whereas the elements of $\boldsymbol{c}$ and d have straightforward equations. We can accomplish these convolutions by using the transformation given in the previous section. A new system is created with the feedback matrix $\boldsymbol{A}^T$ and the input summing vector $\boldsymbol{c}$. This new system has the impulse response $g_i(n)$ at the output of the ith state.

To implement the preceding adaptation equations, the filter structures shown in Fig. 2 can be used to obtain all the required gradients of the system coefficients for a general adaptive state-space filter. The transposed filter with $u(n)$ as its input is used to update the elements of the $\boldsymbol{b}$ vector, whereas each of the other transposed filters is used to adapt the elements of a column of the $\boldsymbol{A}$ matrix. As can be seen, the number of computations required to obtain the gradients for this general state-space filter is quite high: $N+2$ times that of the filter itself.

It is interesting to note that for the specific case where the state-space filter is in the form of the direct-form structure, the gradient equations obtained with this intermediate-function approach are identical to those ob-

[1] These results were for time-invariant lineai systems, whereas the adaptation algorithm makes the overall system nonlinear. The use of these gradients is essentially a linearizing assumption, which is appropriate for the practical case of a small step size, μ.

tained in [6], [7], and thus both approaches result in the same final configuration. This is to be expected, since both methods are finding the derivative of the output with respect to the filter coefficients.

IV. Reduced Computation State-Space Adaptive Filters

To reduce the computations, we note that given an independent set of intermediate functions, $F_i(z)$, any set of desired zeros can be obtained by changing only the $\boldsymbol{c}$ vector and $\boldsymbol{d}$ scalar. This allows one to keep the $\boldsymbol{b}$ vector constant while adapting the remaining state-space system coefficients. Of course, equivalently the $\boldsymbol{c}$ vector could be held constant while the $\boldsymbol{b}$ vector is allowed to change. However, it can be seen from the preceding update equations that the gradient signals required to adapt the $\boldsymbol{c}$ vector are immediately available, whereas obtaining the $\boldsymbol{b}$ vector's gradient signals requires an extra state-space filter. For this reason, we normally choose to adapt the $\boldsymbol{c}$ vector rather than the $\boldsymbol{b}$ vector.

To reduce further the number of computations, we note that N^2 elements of the A matrix are being adapted, where N elements are sufficient to define N poles. Therefore, we look for structures that can be adapted to any set of poles by changing only N elements of the A matrix.

4.1. Single-Column Adaptive Filters

Recall that each of the transposed filters of Fig. 2 provides all the gradients required to adapt a single column of the A matrix. Therefore, if we choose to adapt a modified direct form where only the elements of the last column are adapted, only one transposed filter is required. For the modified direct-form filter, the A matrix has the form

$$A=\begin{bmatrix} 0 & 0 & \cdot & 0 & 0 & a_1 \\ 1 & 0 & \cdot & 0 & 0 & a_2 \\ 0 & 1 & \cdot & 0 & 0 & a_3 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 1 & 0 & a_{N-1} \\ 0 & 0 & \cdot & 0 & 1 & a_N \end{bmatrix}. \tag{21}$$

However, we do not have to restrict ourselves to the modified direct form to obtain computational savings. From control theory, the pole assignment theorem [17] is given next.

Pole Assignment Theorem: The pair $(\boldsymbol{m}^T, A)$ is observable if, and only if, for every complex-conjugate set of N complex numbers there exists a vector $\boldsymbol{k}$ such that the eigenvalues of $(A+\boldsymbol{k}\boldsymbol{m}^T)$ are the given set of N complex numbers.

Thus given an A matrix and choosing $\boldsymbol{m}$ to be the basis vector $\boldsymbol{v}_i$ (a basis vector, $\boldsymbol{v}_i$, consists of all zeros except for the unit element in the ith row), this theorem states that we can obtain any desired set of poles by changing only the ith column of A if $(\boldsymbol{v}_i^T, A)$ is observable (we discuss observability in the next section). Therefore, the poles of an *arbitrary* A matrix can be adapted using only one transposed filter to obtain the necessary gradients required to adapt a single column of A. A block diagram showing how the gradients are obtained for a single-column adaptive filter is shown in Fig. 3.

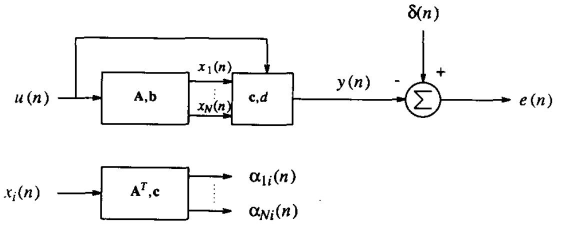


Fig. 3. Generating the gradients for a single-column adaptive filter.

4.2. Column Adaptation Tests

The pole assignment theorem states that $(\boldsymbol{v}_i^T, A)$ must be observable to adapt the ith column. (Here $\boldsymbol{m}$ has been replaced with $\boldsymbol{v}_i$, which restricts discussion to the case of adapting a single column.) To check that this observability constraint is satisfied, the control literature has a number of different tests that could be used. (For a discussion of observability tests, see [18].) Unfortunately, the usual tests are not easily applied to a filter structure with unknown matrix elements. For this reason, as well as to gain some insight, two simple tests are presented herein that allow a designer to check whether the preceding matrix pair or a filter structure satisfies the observability constraint. If the first test is satisfied, then the particular column of A **cannot** be adapted; if the second test is satisfied, then the column **can** be adapted. In most applications, one of these two tests will be satisfied; however, if this is not the case, then one of the many observability tests can be applied.

For both of these tests, consider a given A matrix where one desires to adapt the ith column of A to obtain arbitrary pole locations. Define the vector of functions, $\boldsymbol{G}_{Oi}(z)$, as

$$\boldsymbol{G}_{Oi}^T(z)=\boldsymbol{v}_i^T(z\boldsymbol{I}-A)^{-1}. \tag{22}$$

This vector of functions is visualized most easily as the intermediate $\boldsymbol{G}$ functions of the state-space system having a feedback matrix, A, and an output summing vector, $\boldsymbol{v}_i$; in other words, with the jth element of $\boldsymbol{G}_{Oi}(z)$ being the transfer function from the input of the jth delay operator to the ith state. To derive the two column adaptation tests, an observability independence theorem [18, th. 9-13, p. 432] is used which states the following: The pair $(\boldsymbol{v}_i^T, A)$ is observable if, and only if, the elements of $\boldsymbol{G}_{Oi}(z)$ are linearly independent (over the field of complex numbers).

The two column adaptation tests are:

Column Adaptation Test 1: If any of the elements of $\boldsymbol{G}_{Oi}^T(z)$ is zero, then the ith column of A **cannot** be adapted to arbitrary pole locations.

The proof for this test comes from the fact that the elements of $G_{Oi}(z)$ are not independent if one of the elements is zero. Since the elements of $G_{Oi}(z)$ are not independent, the observability independence theorem implies that the pair (v_i^T, A) is not observable. Therefore, by the pole assignment theorem, the ith column of A cannot be adapted to realize arbitrary poles.

Column Adaptation Test 2: If any of the elements of $G_{Oi}(z)$ is of order N (when canceling poles and zeros are eliminated), then the ith column of A **can** be adapted to realize arbitrary pole locations.

The proof for this test comes from the fact that if any of the elements of $G_{Oi}(z)$ is of order N, then the elements of $G_{Oi}(z)$ are independent. This would not be true for a general set of rational functions, but it is true when they are restricted to be the G functions of an Nth-order system. This fact can be proved by showing that a dependent set of $G_{Oi}(z)$ implies that none of the elements are Nth order. Assume a given feedback matrix A and an output summing vector v_i (sizes $N \times N$ and $N \times 1$, respectively) where the elements of $G_{Oi}(z)$ are dependent. A transposed system consisting of A^T and the input summing vector v_i can be created with the set of intermediate $F_{Oi}(z)$ functions equal to the set of $G_{Oi}(z)$ functions. This transposed system realizes the first equation of the state-space system and uses N delay operators. However, since the elements of $F_{Oi}(z)$ are dependent ($F_{Oi}(z) = G_{Oi}(z)$), one state output can be written as a linear combination of the other states and, thus, one delay operator can be eliminated, leaving the same $F_{Oi}(z)$ functions. However, it is well known that an Nth-order transfer function cannot be created from less than N delay elements and, thus, all the $F_{Oi}(z)$ (and hence $G_{Oi}(z)$) must be less than Nth order. Thus in the case where one of the $G_{Oi}(z)$ elements is Nth order, the set must be independent. Since the elements of $G_{Oi}(z)$ are independent, the observability independence and pole assignment theorems can be applied to prove the stated test.

As an example of applying these tests, consider the fourth-order A matrix:

$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & 0 & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}. \tag{23}$$

This system is a cascade of two biquads—(a_{12}, a_{21}, a_{22}) implement the first biquad and (a_{34}, a_{43}, a_{44}) make up the second biquad, while a_{42} is the feedforward term from the first to the second biquad (see Fig. 4).

Consider the case where one wishes to adapt the first column. It is clear from Fig. 4 that the transfer function from the input of the third or fourth delay operators to the first state is zero. This implies that the last two elements of $G_{O1}(z)$ are zero. Therefore, according to column adaptation test 1, column one cannot be adapted to realize arbitrary poles. This result should come as no surprise, since adapting the first column cannot affect the poles of the second biquad because only feedforward terms are added from the first to the second biquad. In addition, a similar test shows that the second column cannot be adapted to realize arbitrary poles.

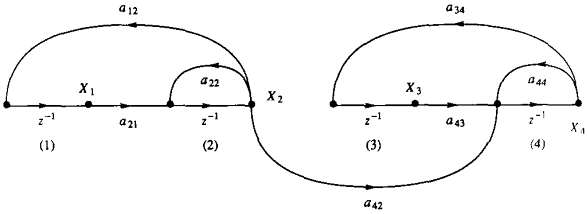


Fig. 4. Signal flow graph for state-space system of cascade of two biquads. Only the A matrix branches are shown.

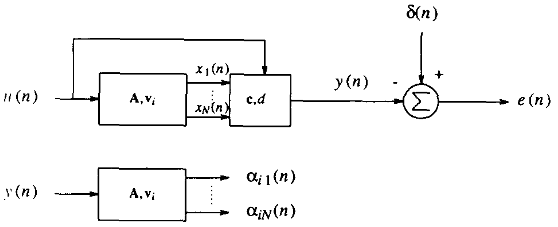


Fig. 5. Generating the gradients for a single-row adaptive filter.

Now, consider the case of adapting the fourth column. In this case, the elements of $G_{O4}(z)$ are the transfer-functions from the inputs of the delay operators to the fourth state. Since the transfer-function from the input of the first delay operator to the fourth state is fourth order, column adaptation test 2 implies that the fourth column can be adapted to give arbitrary poles. Finally, a similar test also shows that the third column can be adapted to realize arbitrary poles.

4.3. *Single-Row State-Space Adaptive Filter*

A situation where only one extra filter is required to obtain the gradients to adapt a single row of the feedback matrix, A, occurs when the input summing vector, b, equals a basis vector, v_i, and the state-space coefficient d is zero. In this case, the transfer-function from the filter input to the filter output is equal to $G_i(z)$. Therefore, to implement (15) for the ith row, only one other system is required that has the functions $F(z)$ at the state outputs. This extra system is created using the b vector and A matrix of the original system. A block diagram showing how to obtain the gradients for a single-row adaptive filter is shown in Fig. 5. Note that if the d coefficient is small, then the gradient signal obtained with this method will be a close approximation to the actual gradient signal.

Of course, to make full use of a single-row adaptive filter where $b = v_i$, it must be possible to adapt the ith row to obtain arbitrary pole positions. To determine whether the ith row is adaptable, one can use the controllability pole assignment theorem [17]. In addition, the column adaptation tests described earlier for adapting a column of the feedback matrix can be modified easily for checking whether a row may be adapted.

Note that in the specific case where the state-space filter is in direct form, the resulting realizations using single-row adaptation are the same as that obtained for direct-form gradient adaptation in [12]. This method of obtaining gradients requires significantly less computations than that originally proposed in [6], [7]. Also note that the nonzero element of the input summing vector does not have to equal 1. If the nonzero element is not unity, then the preceding results still hold but the gradients for the ith row will be scaled. Finally, note that the input vector must have only one nonzero element and the d coefficient must be close to zero for single-row adaptation. These restrictions are not present for single-column adaptation.

4.4. *Choosing the Initial System to Adapt*

Two reduced computation state-space adaptive recursive filters have been described: single-column and single-row structures. However, it is not yet clear what advantages these types of adaptive filters have over the traditional direct-form structure, or how one chooses the initial system to adapt. The advantages of using these new structures will be addressed in this section along with a method of choosing the initial system.

There are two main advantages of using single-row or single-column adaptive filters over direct-form structures. Both of these advantages occur in oversampled applications where final pole locations can be estimated. One advantage is that the final adapted filter could have much better noise performance. This noise improvement can be obtained by choosing a much better structure than the direct form to implement the estimated pole locations. Then, if minor modifications are made to one column, or row, of the feedback matrix, the good noise properties of the structure should be maintained. The second advantage of using these new structures is a much improved convergence rate.

To illustrate the differences in expected convergence rates, we need to consider the error performance surface of the different filter types. Unfortunately, determining the performance surface is not a trivial matter for adaptive IIR filters. However, some insight can be gained by approximating the error performance surface by a quadratic function of the adaptive coefficients (as in [12]) in a close neighborhood of the operating point. As a further simplification, assume that all feedforward coefficients are fixed so that only the performance surface due to feedback coefficients will be investigated. The elements of the correlation matrix, $\boldsymbol{R}$, corresponding to this approximate performance surface can be shown to be

$$R_{ij} = E[\alpha_i \alpha_j] \tag{24}$$

where α_i is the gradient signal used to adapt the ith feedback coefficient. This approximation indicates that one can estimate problems with the performance surface by estimating the degree of correlation between gradient signals. A higher degree of correlation between gradient signals indicates a more ill-conditioned error performance surface and thus slower convergence properties.

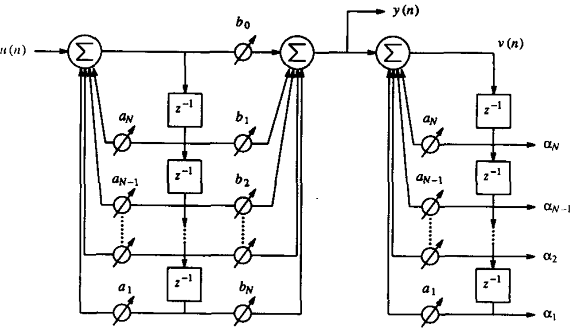


Fig. 6. Direct-form IIR filter and gradient signals.

Consider the direct-form adaptive filter shown in Fig. 6, where the gradient signal used to adapt the a_i coefficient is shown as α_i [12]. Note that although this direct-form adaptive filter requires much less computation to obtain gradient signals than the method proposed in [6], [7], the gradient signals are the same in both cases and, thus, the following reasoning holds for both types of direct-form adaptive filtering. Assuming the coefficients are varying slowly, at an instant of time the transfer-function from the input, $u(n)$, to the filter output, $y(n)$, can be written as

$$\frac{Y(z)}{U(z)} = \frac{N(z)}{D(z)}$$

$$= \frac{b_0 z^N + b_1 z^{N-1} + b_2 z^{N-2} + \cdots + b_N}{z^N - a_N z^{N-1} - a_{N-1} z^{N-2} - \cdots - a_1} \tag{25}$$

where $N(z)$ and $D(z)$ are the numerator and denominator polynomials, respectively. The transfer function from the input, $u(n)$, to the signal $v(n)$ is easily seen to be

$$\frac{V(z)}{U(z)} = \frac{N(z)z^N}{D^2(z)}. \tag{26}$$

In the case of an oversampled low-pass application where $N(z)$ and $D(z)$ make up a low-pass transfer-function with poles clustered around $z = 1$ and the input signal, $u(n)$, has a white noise characteristic, $v(n)$ will be an oversampled low-pass signal. Since the gradient signals α_i are simply delayed versions of this low-pass signal sampled very quickly, it is clear that the gradients, α_i, will be highly correlated. In fact, in the limiting case, as the system oversampling ratio goes to infinity, the gradient signals will be correlated perfectly. Note that this reasoning also predicts that for applications where the signal $v(n)$ continues to have a white noise characteristic, the direct-form IIR filter should adapt quickly. Finally, note that in applications where final pole locations can be estimated, this information is of limited use with a direct-form adaptive filter since there is no obvious method of shaping the performance surface. The only value of estimating final pole locations is in starting the adaptive filter closer to the minimum.

Now, consider the single-row adaptive filter shown in Fig. 5 used in an oversampled low-pass application where estimates of the final pole locations are known. In this case, a state-space system can be designed having its poles equal to the estimated pole locations (only the A and v_i are required) such that the set of state outputs, $x(n)$, is orthonormal for an input signal having a white noise characteristic [19]. This is in contrast to using a direct-form state-space system where the states would be highly correlated in this oversampled application, as discussed earlier. Of equal importance, however, is the fact that in this type of application, the set of states $x(n)$ will have most of its power around the pole locations and thus will remain approximately orthonormal for an input signal that has a relatively flat spectrum response around the pole locations and low power elsewhere. This type of frequency characteristic is the spectrum one would expect for the output signal $y(n)$ in this oversampled application with white noise at the input, $u(n)$. Therefore, for the performance surface around the estimated pole locations, one expects the set of gradient signals, $\alpha_i(n)$, to be approximately orthonormal. This approximately orthonormal set of gradients leads to much improved convergence rates, as will be shown in some simulation results presented in Section VI.

Unfortunately, it is not as simple to make a qualitative argument concerning the performance of the single-column adaptive filter. However, it has been observed that using the same method of choosing the nonadaptive elements as in the single-row case and adapting the last column, good adaptation performance has resulted. Some examples of simulation results for the single-column adaptive filter are also given in Section VI.

Finally, it should be mentioned that the design of orthonormal filters for arbitrary pole locations often results in dense matrices; therefore, this paper will make use of a variation of an *orthonormal ladder filter* structure presented in [20]. (For design details, see the Appendix.) We shall refer to realizations obtained with this approach as "quasi-orthonormal filters" since the resulting realizations approach true orthonormal filters as the ratio of the sampling frequency to the passband edge is increased. Of course, one is not restricted to using the quasi-orthonormal structure, and the authors are currently investigating the effect of using other state-space structures.

V. Roundoff Noise Comparison

In this section, the possible noise performance improvement of using a single-row adaptive filter over a direct-form adaptive filter will be demonstrated through the use of an example.

First, a measure for comparing the noise performance of different filter structures needs to be defined. Our noise measure, N_M, is a slight variant of the measures presented in [21], [22],

$$N_M = \mathrm{trace}(\boldsymbol{KWQ}) \tag{27}$$

where $\boldsymbol{K}$ and $\boldsymbol{W}$ are defined as

$$\boldsymbol{K} = \boldsymbol{AKA}^T + \boldsymbol{bb}^T = \sum_{k=0}^{\infty} (\boldsymbol{A}^k\boldsymbol{b})(\boldsymbol{A}^k\boldsymbol{b})^T \tag{28}$$

$$\boldsymbol{W} = \boldsymbol{A}^T\boldsymbol{WA} + \boldsymbol{c}^T\boldsymbol{c} = \sum_{k=0}^{\infty} (\boldsymbol{cA}^k)^T(\boldsymbol{cA}^k) \tag{29}$$

and $\boldsymbol{Q}$ is a diagonal matrix, where Q_{ii} is zero if the elements of row i of $\boldsymbol{A}$ consist only of 0's, 1's, and -1's. Otherwise, Q_{ii} is 1.

Note that this noise measure is valid when using a modern digital signal processor that has a multiplier/accumulator that does not truncate until writing out to memory. Thus, the noise model used assumes each row of a state-space system has one noise source due to truncation error rather than a noise source for each nonzero element. The matrix $\boldsymbol{Q}$ makes an adjustment for rows where no truncation errors are introduced. In the case of a direct-form filter, there are $N-1$ rows where no truncation errors are present.

With a noise measure to compare different filter structures, we may now proceed with an example. The example used will be an oversampled transfer-function, typical in many practical applications. It is well known that direct-form filters have poor noise performance for such applications. The example transfer function is that of a third-order low-pass filter with a sampling frequency/passband frequency ratio of about 32. The same transfer function is used in the simulation results of Section VI; the poles and zeros are given in the last row of Table I in Section VI. Three different filter realizations are investigated with respect to noise performance.

The first realization is of the direct-form type, having the following state-space system description:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.8889 & -2.7432 & 2.8523 \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\boldsymbol{c}^T = [0.01003 \quad -0.01884 \quad 0.01088] \quad d = 0.005312. \tag{30}$$

The $\boldsymbol{K}$ and $\boldsymbol{W}$ matrices for the direct-form realization are

$$\boldsymbol{K} = \begin{bmatrix} 13\,522 & 13\,395 & 13\,017 \\ 13\,395 & 13\,522 & 13\,395 \\ 13\,017 & 13\,395 & 13\,522 \end{bmatrix} \tag{31}$$

$$\boldsymbol{W} = \begin{bmatrix} 0.0382 & -0.0800 & 0.0426 \\ -0.0800 & 0.1679 & -0.0898 \\ 0.0426 & -0.0898 & 0.0483 \end{bmatrix}. \tag{32}$$

One can easily calculate the noise measure, N_M, for this filter to be 652.

The next two filter realizations are obtained using the quasi-orthonormal filter structure described in the Appendix.

Implementing the described oversampled transfer function by a quasi-orthonormal filter, the following state-space system is obtained:

$$A=\begin{bmatrix}1 & 0.1188 & 0\\ -0.1188 & 1 & 0.1567\\ 0 & -0.1567 & 0.8523\end{bmatrix}\quad b=\begin{bmatrix}0\\0\\0.2168\end{bmatrix}$$

$$c^T=[0.4755\quad 0.0859\quad 0.2168]\quad d=0.005312. \tag{33}$$

The $\boldsymbol{K}$ and $\boldsymbol{W}$ matrices for this structure are

$$K=\begin{bmatrix}0.2202 & -0.01746 & -0.05179\\ -0.01746 & 0.2941 & -0.04854\\ -0.05179 & -0.04854 & 0.2456\end{bmatrix} \tag{34}$$

$$W=\begin{bmatrix}1.6698 & 1.0369 & 0.4573\\ 1.0369 & 1.4300 & 0.9107\\ 0.4573 & 0.9107 & 1.02656\end{bmatrix}. \tag{35}$$

Note that the $\boldsymbol{K}$ matrix indicates the structure is close to being orthonormal. The noise measure N_M, for this filter is 1.04, which is not much worse than the Mullis and Robert's optimal filter [21] that has a noise measure of 0.73. (The noise measure for the optimal filter can be calculated from the eigenvalues of $\boldsymbol{KW}$.) Thus the quasi-orthonormal design approach appears to be a good structure for oversampled filters. Of course, this low-noise measure of 1.04 would be obtained only if the exact pole locations were known, a situation where an adaptive filter is not required.

We now proceed to investigate the case where a single row of the feedback matrix A is adapted to move the poles from an initial position to their final position. Using the same design procedure as described earlier, a quasi-orthonormal state-space realization was obtained with all its poles at 0.9. The poles of this filter were then adapted to the pole locations of the desired transfer function by changing only the last row of the A matrix. The $\boldsymbol{c}$ vector and d scalar were also changed to obtain the desired zeros. The following state-space system was obtained.

$$A=\begin{bmatrix}1 & 0.0577 & 0\\ -0.0577 & 1 & 0.1633\\ -0.1686 & -0.2163 & 0.8524\end{bmatrix}\quad b=\begin{bmatrix}0\\0\\0.309\end{bmatrix}$$

$$c^T=[0.6984\quad 0.0579\quad 0.0352]\quad d=0.00531. \tag{36}$$

Finally, the following $\boldsymbol{K}$ and $\boldsymbol{W}$ were found for the preceding system.

$$K=\begin{bmatrix}0.1149 & -0.0187 & -0.1846\\ -0.0187 & 0.6493 & -0.0814\\ -0.1846 & -0.0814 & 0.7709\end{bmatrix} \tag{37}$$

$$W=\begin{bmatrix}4.8721 & 1.4571 & 1.0212\\ 1.4571 & 0.6479 & 0.4300\\ 1.0212 & 0.4300 & 0.5054\end{bmatrix}. \tag{38}$$

The noise measure, N_M, for this filter is 1.4, which is slightly worse than the preceding case but is still orders of magnitude better than the figure for the direct-form implementation. Thus, it appears that low roundoff noise adaptive filters may be obtained if one has a good estimate of the final pole locations. Of course, if the starting pole locations are farther from the final pole locations, one would expect a higher noise figure. Other examples of roundoff noise improvement will be presented in the next section.

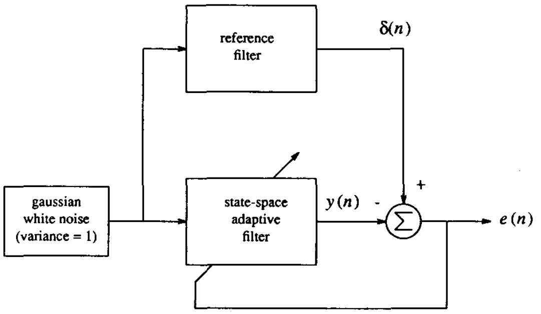

reference filter
δ(n)
gaussian white noise (variance = 1)
state-space adaptive filter
y(n)
-
+
Σ
e(n)

Fig. 7. System identification application used for simulations.

VI. Simulation Results

This section will present computer simulation results of the preceding structures used as adaptive filters. The simulations are based on system identification applications where an Nth-order adaptive filter is required to adjust its coefficients to match an Nth-order reference transfer function. A block diagram of the application used for these simulations is shown in Fig. 7. For convenience in calculation and in keeping with the conventions of the literature, we monitor coefficient convergence. Note that in the general case, for instance, when there is a pole-zero cancellation in the reference filter, mean-squared error minimization does not necessarily imply coefficient convergence.

To illustrate the advantage of the new structures for oversampled applications, the reference filter will have varying ratios of sampling frequency to passband edge frequency. All the reference filters are derived from a third-order elliptic low-pass analog prototype with the following s-plane poles and zeros.

$$\text{poles}=\{-0.3226,\,-0.1343\pm j0.91920\}$$
$$\text{zeros}=\{\pm j2.2705,\infty\}. \tag{39}$$

The passband of the prototype has a 3-dB ripple with the passband edge normalized to 1 rad/s.

To obtain oversampled digital filters with varying bandwidths, the bilinear transform [23] was applied to the analog prototype

$$z=\frac{1+(T/2)s}{1-(T/2)s} \tag{40}$$

where T is the sampling period. Using the fact that the analog prototype's passband edge is normalized, one can use the well-known prewarping equation to find a relationship between the sampling period, T, and the ratio of the sampling frequency, ω_s, to the passband frequency, ω_p. For purposes of comparison, four values of the sampling period, T, are used: 2, 0.8, 0.4, and 0.2. These correspond to ratios of sampling frequency to passband edge frequency of approximately 4, 8, 16, and 32, respectively. In all cases, the digital transfer functions were

TABLE I
ADAPTATION RATES AND NOISE MEASURES FOR FILTERS OF VARYING BANDWIDTHS

$\frac{\omega_s}{\omega_p}$	Transfer Function	Initial Poles		Direct Adapt	Row Adapt	Column Adapt
4	poles 0.5122 $0.06429 \pm j0.8625$	0	Step Size μ	0.01	0.0025	0.0028
		0	Iterations for Convergence	16K	60K	50K
	zeros -1.0 $-0.6751 \pm j0.7377$	0	Noise Measure N_M	0.8	8.2	8.2
8	poles 0.7714 $0.6920 \pm j0.5904$	0.7	Step Size μ	0.00028	0.03	0.015
		0.7	Iterations for Convergence	500K	60K	35K
	zeros -1.0 $0.09597 \pm j0.9954$	0.7	Noise Measure N_M	4.8	2.4	5.0
16	poles 0.8788 $0.8872 \pm j0.3379$	0.8	Step Size μ	0.000025	0.01	0.00375
		0.8	Iterations for Convergence	6MEG	50K	30K
	zeros -1.0 $0.6581 \pm j0.7529$	0.8	Noise Measure N_M	49.	1.6	2.8
32	poles 0.9375 $0.9574 \pm j0.1775$	0.9	Step Size μ	0.000001	0.0125	0.0015
		0.9	Iterations for Convergence	>10MEG	40K	40K
	zeros -1.0 $0.9019 \pm j0.4318$	0.9	Noise Measure N_M	652.	1.4	2.6

scaled to have a gain of 1 at $z = 1$. For each of these values of T, three different structures for the adaptive filter are used: direct-form, single-row, and single-column adaptive filters.[2] The single-row and single-column adaptive filters start from the quasi-orthonormal structure and then either the last row or column is adapted. The initial pole locations of the adaptive filters are three coincident poles on the real axis at a point chosen close to the final pole locations. In all three cases, the initial pole locations are the same, and the c vector and d scalar are both set to zero.

Of course, the step size μ is an important factor in controlling the adaptation rate. Therefore, a method is required for choosing the step size for each simulation so that a fair comparison can be made. First, it should be pointed out that the same step size was used for all the state-space elements and no power normalization was used. To find the step size for a particular simulation, first, a trial-and-error method was used to find a "diverging step size" that caused the simulation to go unstable *after* 500 iterations. This diverging step size appeared to vary by at most 20 percent for a particular simulation. The step size then used for simulations was the diverging step-size value divided by 4. One performance measure used is the "iteration for convergence," taken as the number of iterations required to have the coefficients of the state-space system converge to four significant digits.

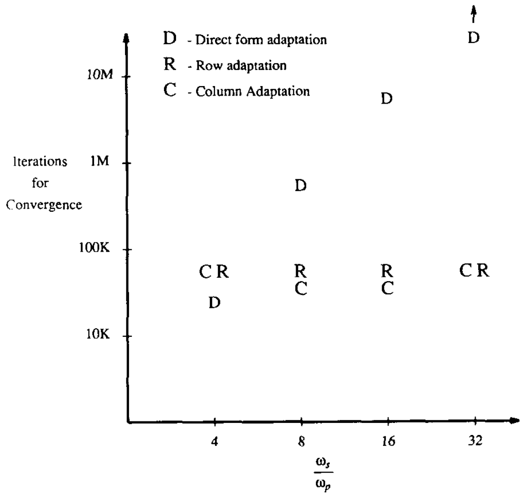


Fig. 8. Convergence times of different structures for filters of varying bandwidths.

Table I lists the results of the different simulations. Note that as the reference filter becomes more oversampled, the direct form takes much longer to adapt than either of the other two structures. Also note that the noise measure, N_M, of the final adapted filter is higher for the direct-form case than the other structures in the cases of the high sampling frequency/passband edge ratio. In the case of the lowest ratio, the noise measure of the row and column adaptation structures is relatively poor because the quasi-orthonormal filter has poor noise properties at pole locations far from $z = 1$. The graph in Fig. 8 summarizes the convergence times for the varying oversampled reference filters and different adaptive filter structures. These results indicate that using structures other than the direct form can result in much better adaptation rates in oversampled applications. Note that while these results for the third-order case show a dramatic difference between the single-row (or column) filter and the direct-form structure, it is likely that this difference will be even greater for higher order filters.

To see why the direct-form filter performs so poorly at high oversampled rates, we now compare the performance surfaces corresponding to the simulation results presented in the last row of Table I. However, whereas all the coefficients in the bottom row of A and all the coefficients in c and d were varied in the simulation results of the previous section, to obtain a contour plot of a performance surface, we arbitrarily choose to vary only two coefficients, A_{31} and A_{32}. All the remaining coefficients are fixed to their optimum values such that the error signal can go to zero. This is equivalent to viewing a cross section of the full performance surfaces for the simulation results of the last row of Table I.

Fig. 9 shows contour plots of the cross section of the performance surfaces for both the single-row and direct-form filters. It should be mentioned that the corresponding heights for both plots are the same, but the axis scale

[2] For the single-row adaptive filter, it should be pointed out that the state-space d element is close to zero and, thus, an approximate gradient is used.

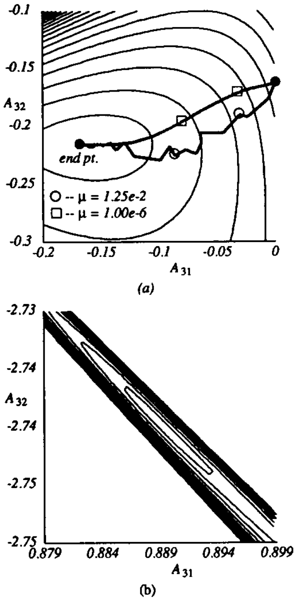


Fig. 9. Examples of performance surface cross sections for different filter structures. (a) Single-row structure. Also shown are adaptation paths for small ($\mu = 1.00e-6$) and large ($\mu = 1.25e^{-2}$) stepsize. Note that for a small stepsize, the adaptation path follows the path of steepest descent, as expected. (b) Direct-form structure (view expanded 10 times).

for the direct-form filter is ten times smaller than that for the single-row filter. This factor of 10 implies that the direct-form filter's surface is even more ill-conditioned than that shown in Fig. 9(b). Comparing the two performance surfaces of Fig. 9, it is clear that since the direct-form filter's surface is much more ill-conditioned than the single row's surface, one expects the single-row structure to adapt faster, as was shown in the simulation results. Note that in Fig. 9(a), the adaptation paths for the single-row filter are also shown, demonstrating that for small step sizes, the adaptation path does indeed follow the path of steepest descent. A larger step-size adaptation path is also shown.

Before concluding this section, it is of interest to compare the total amount of computations for the single-row filter with respect to the direct-form structure. Here, we shall use the number of multiplications by coefficients other than one or zero as a measure of computational complexity. Since the two structures each have $2N$ coefficients that are being varied, the number of multiplies per iteration for the LMS update equations will be the same. However, to obtain the necessary gradient signals, the total number of multiplies will be different for the two structures. As seen for the direct-form filter in Fig. 6, the number of multiplications per iteration to obtain the gradient signals is $3N+1$. Assuming the single-row filter uses a quasi-orthonormal structure as the initial system, it is not difficult to show that the total number of multiplies per iteration to obtain the gradient signals is $7N-5$. Therefore, for the preceding third-order case, the single-row filter requires 16 multiplications, whereas the direct-form structure requires 10 multiplications, a factor of only 1.6 difference. However, it is clear from the simulation results that the total computations to convergence for the single-row filter will be much smaller than that of the direct-form filter in the oversampled case. It should be mentioned that the single-column filter will use the same number of computations per iteration as the single-row filter.

VII. Conclusions

An algorithm was presented for adapting a general state-space filter. This algorithm required $N+2$ state-space filters to obtain all the gradients required for adapting an Nth-order system. Single-column and single-row adaptive filter structures then were introduced where only two state-space filters are required to obtain the necessary gradients. It was shown that in applications where a good estimate of the final pole locations is known, single-column or single-row adaptive filters can result in improved convergence rates and significantly better round-off noise performance as compared with direct-form implementations. These new adaptive filters are especially effective in the practical case of oversampled systems where the adaptive filter is used to "fine-tune" a transfer-function.

It should be mentioned here that all the gradient signals required for the adaptation methods proposed in this paper can be obtained as the outputs of filters. Thus the algorithms presented are applicable in both the discrete- and continuous-time domains. Although this paper presents theory concerning digital adaptive filters, the authors are currently investigating fully integrated analog adaptive filters. As a final comment, it should be mentioned that since the presented algorithms are all gradient-based using IIR structures, the issues involving filter instability and global convergence should be solved before these techniques become useful realizations.

Appendix
Quasi-Orthonormal Design Procedure

A state-space orthonormal design technique was presented for continuous-time circuits [20]. The structures resulting from this technique have the advantages that they are inherently L_2 scaled for dynamic range and have good sensitivity and noise performance. As well, the feedback matrix is nearly skew-symmetric and sparse. This final property is particularly interesting since an orthonormal digital filter is usually dense. In this section, we present a procedure to obtain, for oversampled transfer functions, a nearly orthonormal state-space digital filter with a sparse feedback matrix.

The design uses the fact that the forward difference transformation applied to a state-space system simply shifts poles and zeros by $+1$ and changes the feedback matrix by adding one to each of the diagonal elements. Specifically, given a state-space system

$$\begin{aligned} sx &= Ax + bu \\ y &= c^T x + du \end{aligned} \qquad \text{(A.1)}$$

if the forward difference transformation $s = z - 1$ is applied, the following system is obtained:

$$\begin{aligned} zx &= (A + I)x + bu \\ y &= c^T x + du \end{aligned} \tag{A.2}$$

where the poles and zeros in the z plane are simply shifted versions of the poles and zeros in the s plane.

Using this transformation property, the quasi-orthonormal design procedure is as follows.

1) Shift both poles and zeros in the z plane by -1.
2) Obtain an orthonormal state-space system for the shifted poles and zeros using the approach in [20] for continuous-time designs.
3) Shift the poles and zeros of the state-space system by $+1$ by adding one to each of the diagonal elements of the feedback matrix A.

Note that this design technique is exact in the sense that it produces exactly the desired transfer function; however, it does not exactly reproduce the orthonormal states of the continuous-time filter. With this approach, the resulting filters approach orthonormal behavior as the ratio of the sampling frequency to passband edge is increased. Specifically, the diagonal elements of K will become asymptotically equal and the off-diagonal elements will approach zero. It should be noted, however, that the diagonal elements will be a factor of 2π less than unity. This factor arises because unit-variance white noise in discrete-time systems spreads noise power over the 2π circumference of the unit circle, whereas noise in continuous-time systems is defined as having unit power density over 1 rad/s. Thus a quasi-orthonormal system obtained as described earlier will asymptotically have mean-square output levels a factor of 2π below the mean-square variance of the discrete-time white input. Of course, a simple scaling of the input vector can be used to obtain an arbitrary mean-square level. Finally, note that this design method always will result in a sparse tridiagonal structure for the A matrix.

Acknowledgment

The authors would like to thank B. A. Francis for invaluable discussions during the development of this paper.

References

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

[2] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 939–947, Oct. 1986.

[3] P. L. Feintuch, "An adaptive recursive LMS filter," *Proc. IEEE*, vol. 64, pp. 1622–1624, Nov. 1976.

[4] T. C. Hsia, "A simplified adaptive recursive filter design," *Proc. IEEE*, vol. 69, pp. 1153–1155, Sept. 1981.

[5] M. G. Larimore, J. R. Treichler, and C. R. Johnson, "SHARF: An algorithm for adapting IIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 428–440, Aug. 1980.

[6] S. D. Stearns, G. R. Elliott, and N. Ahmed, "On adaptive recursive filtering," in *Proc. 10th Asilomar Conf. Circuits Syst. Comput.*, 1976, pp. 5–10.

[7] S. A. White, "An adaptive recursive digital filter," in *Proc. 9th Asilomar Conf. Circuits Syst. Comput.*, 1975, pp. 21–25.

[8] I. L. Ayala, "On a new adaptive lattice algorithm for recursive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 316–319, Apr. 1982.

[9] D. Parikh, N. Ahmed, and S. D. Stearns, "An adaptive lattice algorithm for recursive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 110–112, Feb. 1980.

[10] K. W. Martin and M. T. Sun, "Adaptive filters suitable for real-time spectral analysis," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 218–229, Feb. 1986.

[11] M. Nayeri and W. K. Jenkins, "Analysis of alternate realizations of adaptive IIR filters," in *1988 IEEE Int. Symp. Circuits and Systems*, pp. 2157–2160.

[12] F. F. Yassa, "Optimality in the choice of the convergence factor for gradient-based adaptive algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 48–59, Jan. 1987.

[13] T. Söderström and P. Stoica, "Some properties of the output error method," *Automatica*, vol. 18, pp. 93–99, Jan. 1982.

[14] H. Fan and M. Nayeri, "On error surfaces of sufficient order adaptive IIR filters: Proofs and counterexamples to a unimodality conjecture," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 1436–1442, Sept. 1989.

[15] W. M. Snelgrove and A. S. Sedra, "Synthesis and analysis of state-space active filters using intermediate transfer functions," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 287–301, Mar. 1986.

[16] L. B. Jackson, "On the interaction of roundoff noise and dynamic range in digital filters," *Bell Syst. Tech. J.*, vol. 49, pp. 159–184, 1970.

[17] W. M. Wonham, *Linear Multivariable Control.* New York: Springer-Verlag, 1974.

[18] B. C. Kuo, *Digital Control Systems.* New York: Holt, Rinehart and Winston, 1980.

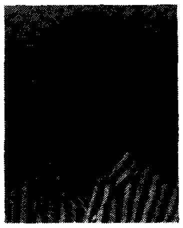
[19] R. A. Roberts and C. T. Mullis, *Digital Signal Processing.* Reading, MA: Addison-Wesley, 1987.

[20] D. A. Johns, W. M. Snelgrove, and A. S. Sedra, "Orthonormal ladder filters," *IEEE Trans. Circuits Syst.*, vol. CAS-36, pp. 337–343, 1989.

[21] C. T. Mullis and R. A. Roberts, "Synthesis of minimum roundoff noise fixed point digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp. 551–562, 1976.

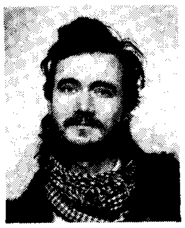
[22] G. Amit and U. Shaked, "Small roundoff noise realization of fixed-point digital filters and controllers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 880–891, June 1988.

[23] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.



David A. Johns (S'81–M'89) received the B.A.Sc., M.A.Sc., and Ph.D. degrees from the University of Toronto, Canada, in 1980, 1983, and 1989, respectively, all in electrical engineering.

From 1980 to 1981, he worked as an Applications Engineer in the Semiconductor Division of Mitel Corporation, Ottawa, Canada. From 1983 to 1985, he was an analog IC designer at Pacific Microcircuits Ltd., Vancouver, Canada. Upon completion of his doctoral work, he joined the University of Toronto, where he is currently an Assistant Professor. His research interests are in the areas of adaptive systems and integrated analog circuit design.



W. Martin Snelgrove (S'75–M'81) received the B.A.Sc. in chemical engineering and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, in 1975, 1977, and 1982, respectively.

In 1982, he worked at the Instituto Nacional de Astrofisica, Optica y Electronica in Mexico as a Visiting Researcher in CAD. Since then, he has been at the University of Toronto, where he is currently an Associate Professor. He is active in multiprocessor CAD and integrated signal-processing systems, and particularly in the interface between the CAD and circuit areas. A 1986 paper coauthored with A. Sedra was the winner of the 1986 Circuits and Systems Society Guillemin–Cauer Award.

Adel S. Sedra (M'66–SM'82–F'84) received the B.Sc. degree from Cairo University, Egypt, in 1964, and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Canada, in 1968 and 1969, respectively, all in electrical engineering.

Since 1969, he has been on the staff of the University of Toronto, where he is currently Professor and Chairman of the Department of Electrical Engineering. He also serves as an industrial consultant and is one of the founders and a director of the Information Technology Research Centre, a center of excellence funded by the Province of Ontario. His research work has been in the area of microelectronic filter design. He has published about a hundred papers and two books.

Dr. Sedra's papers have won two IEEE awards, and, in 1988, he received the ASEE Frederick Emmons Terman Award for excellence in engineering education.