Continuous-Time LMS Adaptive Recursive Filters

David A. Johns, *Member, IEEE*, W. Martin Snelgrove, *Member, IEEE*, and Adel S. Sedra, *Fellow, IEEE*

***Abstract*—This paper presents an approach for implementing continuous-time adaptive recursive filters. The resulting filters should be capable of operating on much higher signal frequencies than their digital counterparts since no sampling is required. With respect to implementation problems, the effects of dc offsets are investigated and formulas derived so that these effects can be estimated and reduced. As well, it is shown that the dc offset performance is strongly affected by the choice of structure for the adaptive filter. Finally, experimental results from a discrete prototype are given where accurate adaptation is observed and dc offset effects are compared to theoretical predictions.**

I. Introduction

RECENTLY, the hardware efficient least-mean-squared (LMS) algorithm was applied to state-space adaptive recursive filters in digital form [1]. It is of interest to modify this adaptive digital approach for use with *continuous-time* adaptive filters since, in many applications, analog circuits have definite advantages over their digital counterparts. For example, analog circuits are often used in applications requiring high-frequency signal-processing capability and/or ones that require small integrated-circuit area. With this motivation, this paper presents a design methodology for the realization of continuous-time adaptive recursive filters using the LMS algorithm. (For an overview of adaptive recursive filters in the digital realm, the reader is referred to [2].)

With any adaptive filter, there is a requirement for a programmable filter. This requirement is one of the reasons that digital-signal-processing has dominated adaptive filter implementations since most digital filters are inherently programmable by modifying memory locations. In the analog realm, programmable filters can be created by making use of one of the many techniques that have been proposed for realizing integrated continuous-time filters [3]. To account for process and temperature variations, these integrated filters have the ability to tune integrator time constants implying an inherent programming method that adjusts filter coefficients. This inherent programmability of integrated continuous-time filters was a strong motivation for developing the work presented in this paper. As an additional benefit, a recent publication shows that the work presented here can also be used as a basis for tuning integrated continuous-time filters [4].

Previous work related to adapting poles in continuous-time filters can be traced back to solving optimization problems with analog computers in the 1960's [5], [6]. By making use of gradient signals together with the LMS algorithm, this analog computer approach results in identical block diagrams to those presented in this paper. However, one major problem with this previous approach is that it is only applicable to filters based on the direct-form structure. The approach presented in this paper extends that basic method to filters with arbitrary structures. It should be mentioned here that it has been shown that the choice of structures can significantly affect the performance of the overall system in the digital realm [1] and it will be shown in this paper that dc offset effects are also greatly affected by structure choice.

In more recent work on continuous-time adaptive recursive filters, experimental results were given for a second-order adaptive filter using the sequential-linear-search (SLS) algorithm [7], [8]. The SLS algorithm is similar to the LMS algorithm in that a steepest descent search is performed to locate a minimum in the performance surface. However, rather than using gradient signals, the gradient is estimated by changing a filter coefficient and then observing the direction of change in the mean-squared value of the error signal. The problems with this techniques are that only one coefficient can be adjusted at a time and more importantly, it is difficult to obtain an accurate measurement of a deviation in the mean-squared error signal.

The outline of this paper is as follows. A general approach for continuous-time LMS adaptive recursive filters applicable to arbitrary filter structures is presented in Section II. In Section III, it is shown how this approach is applied to state-space systems and, in particular, a single-row adaptive filter. In Section IV, the effects of dc offsets are discussed and formulas developed to predict these effects. As well, a simple method to reduce the dc offset effects is presented. Finally, experimental results of a discrete single-row adaptive filter prototype are presented in Section V showing the practicality of the ap-

Manuscript received June 21, 1990; revised February 12, 1991. This paper was recommended by Associate Editor D. Graupe.

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

IEEE Log Number 9144830.

0098-4094/91/0700-0769$01.00 ©1991 IEEE

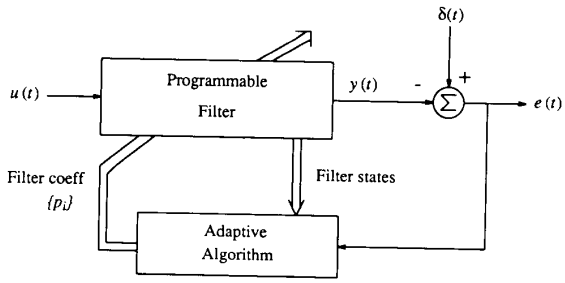


Fig. 1. Block diagram of an adaptive filter.

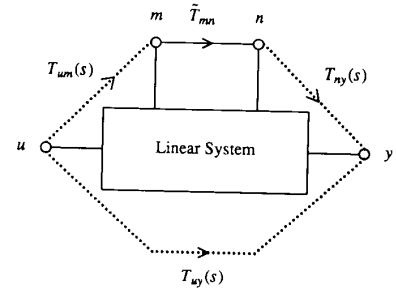


Fig. 2. Signal-flow-graph where the arm $\tilde{T}_{mn}$ is shown separately from the rest of the system.

proach and comparing experimental and theoretical results for dc offset effects.

II. LMS Adaptive Recursive Filters

A block diagram of an adaptive filter is shown in Fig. 1. The system has two inputs—the filter input, $u(t)$, and the reference signal, $\delta(t)$. As well, an error signal, $e(t)$, is created as the difference between the programmable filter's output, $y(t)$, and the reference signal. Describing the LMS algorithm qualitatively, the goal of the adaptive filter is to minimize the mean-squared value of the error signal by slowly adjusting (in comparison to the signal frequency) the transfer function[1] of the programmable filter through the adjustments of the filter coefficients, p_i. In other words, defining an error performance surface as the mean-squared error value for varying filter coefficients, the LMS algorithm results in a steepest descent search to find a minimum in that surface.

By minimizing the mean-squared error value, adaptive filters can be used in a variety of applications such as channel equalization, noise cancellation, and others [9]. However, one application worth noting is the "model-matching" application as it is often used for testing purposes. In this application, a sufficiently exciting input signal, $u(t)$, is injected into the adaptive filter as well as to an external reference filter. The reference signal, $\delta(t)$, is then taken as the output of the reference filter. With this setup, after adaptation, the transfer function of the programmable filter should match that of the reference filter or be some approximation of it. It should be mentioned here that for the purposes of this paper, we assume the input signal and reference filter to be stationary. Although nonstationary inputs are the usual case in most applications and their effects can sometimes dictate the performance of LMS filters, the performance degradation for these types of inputs are beyond the scope of this paper.

To derive the LMS algorithm, we first look at the discrete-time case where a steepest descent algorithm in the mean-squared error performance surface can be implemented by realizing the following equation.

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

Here, $E[\bullet]$ denotes expectation and μ is a small positive parameter that controls the rate of convergence. Since the partial derivative of the mean-squared error signal is not practical to obtain, the instantaneous squared error is used as an approximation of the mean-squared error resulting in the LMS algorithm. Making this substitution and using the fact that $e(n) = \delta(n) - y(n)$, the following LMS update equation is obtained [9].

$$p_i(n+1) = p_i(n) + 2\mu e(n)\phi_i(n). \tag{2}$$

In the above equation, $\phi_i(n)$ is a gradient signal defined as

$$\phi_i(n) = \left.\frac{\partial y(n)}{\partial p_i}\right|_{p_i = p_i(n)}. \tag{3}$$

Note that although the instantaneous gradient may often indicate the wrong direction, the average value of the gradient will be correct.

This discrete-time algorithm can be extended into the analog domain quite naturally as [10]

$$p_i(t) = 2\mu \int_0^t e(\tau)\phi_i(\tau)\, d\tau \tag{4}$$

where $\phi_i(\tau)$ is the equivalent analog gradient signal. Note that when $p_i(t)$ becomes a constant value at a minimum in the performance surface, the average value of the input to the integrator must be zero. Since the average value of the input to the integrator is defined as the correlation between the error and gradient signals, this simple observation explains the well-known fact that the gradient signals must be uncorrelated with the error signal after adaptation is successfully completed.

It is seen from the above adaptation equation that although the signal $e(t)$ is readily available, the gradient signal, $\phi_i(t)$ must be obtained. To obtain these gradient signals, one can make use of a sensitivity formula that relates the change in the output signal to a change in a single arm of a signal-flow-graph [11], [12]. Specifically, consider the system represented by the signal-flow-graph in Fig. 2 where $\tilde{T}_{mn}$ is a filter coefficient in the linear

[1]Although the use of linear transfer functions does not strictly apply to adaptive systems since they are nonlinear systems, this idea of varying transfer functions is essentially a linearizing assumption that is appropriate for the practical case of slow adaptation.

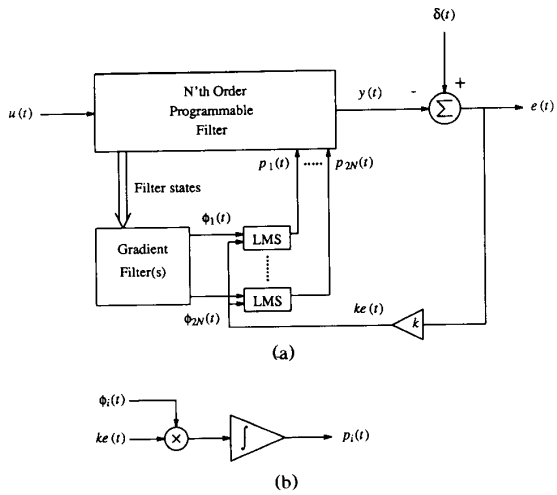


Fig. 3. Block diagram of a LMS gradient adaptive filter. (a) Overall system. (b) Details of the LMS block.

system (shown separately) and $T_{ij}(s)$ is the transfer function from node i to node j. For this system, it can be shown that

$$\frac{\partial T_{uy}(s)}{\partial \tilde{T}_{mn}} = T_{um}(s)T_{ny}(s). \tag{5}$$

Since $U(s)$ does not change as $\tilde{T}_{mn}$ changes, we can also write

$$\frac{\partial Y(s)}{\partial \tilde{T}_{mn}} = T_{um}(s)T_{ny}(s)U(s). \tag{6}$$

In the time domain, this sensitivity formula becomes

$$\frac{\partial y(t)}{\partial \tilde{T}_{mn}} = t_{um}(t) \otimes t_{ny}(t) \otimes u(t) \tag{7}$$

where the symbol $\otimes$ denotes convolution and $t_{ij}(t)$ is the impulse response of the transfer function $T_{ij}(s)$. This sensitivity equation states that the necessary gradient signals can be obtained for arbitrary linear systems by applying the input signal to a cascade of systems with the proper transfer functions. This approach (also used in the digital domain [13]) leads to the general block diagram shown in Fig. 3. It should be mentioned here that the choice of structure for the programmable filter determines the complexity of the gradient filter(s) as well as the performance of the adaptive system [1]. Also of interest is the gain block, k, which as we shall see, is introduced to reduce the effects of dc offsets. Finally, for simplicity, it is assumed that the ith filter coefficient is a linear function of the controlling voltage, p_i, created at the output of the coefficient update integrator in Fig. 3(b). In fact, for an analog programmable filter, the ith coefficient will most likely be some monotonic nonlinear function of the controlling input voltage. However, it is not difficult to see that this nonlinear function will not affect the steady-state values of the coefficients.

III. State-Space Adaptive Filters

Although the above approach can be used for arbitrary programmable filters, it is not always clear how much extra circuitry will be required to obtain the necessary gradient signals or which filter coefficients should be allowed to change in order to obtain arbitrary poles and zeros while keeping the number of filter coefficients to a minimum. For these reasons, we choose to implement the programmable filter as a linear system that has a one-to-one correspondence with state-space systems for which these issues are mathematically well understood. It should be emphasized here that most filter structures (e.g., direct-form, cascade, ladder-simulation, etc.) are special cases of state-space filters.[2]

An Nth-order state-space linear time-invariant system is described by the following equations:

$$\begin{aligned} s\boldsymbol{X}(s) &= \boldsymbol{A}\boldsymbol{X}(s) + \boldsymbol{b}U(s) \\ Y(s) &= \boldsymbol{c}^T\boldsymbol{X}(s) + dU(s) \end{aligned} \tag{8}$$

where $U(s)$ is the input signal; $\boldsymbol{X}(s)$ is a vector of N states, which in fact are the integrator outputs; $Y(s)$ is the output signal; and $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and d are coefficients relating these variables. The transfer-function of the above system is easily shown to be

$$T(s) = \boldsymbol{c}^T(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} + d. \tag{9}$$

From (9), we can see that the poles of the system are the eigenvalues of $\boldsymbol{A}$ whereas the zeros of the system are related to all four of the system coefficients. It should be mentioned here that not all N^2 coefficients of $\boldsymbol{A}$ should be adapted independently. In fact, the number of degrees of freedom in the coefficients of $\boldsymbol{A}$ should equal the filter order; otherwise, coefficient drift or improper adaptation might occur.

To obtain the gradient signals, the approach described in [1] can be easily modified for continuous-time use resulting in the following coefficient-update formulas for the state-space coefficients.

$$A_{ij}(t) = 2\mu \int_0^t e(\tau)\alpha_{ij}(\tau)\, d\tau \tag{10}$$

$$b_i(t) = 2\mu \int_0^t e(\tau)\beta_i(\tau)\, d\tau \tag{11}$$

$$c_i(t) = 2\mu \int_0^t e(\tau)x_i(\tau)\, d\tau \tag{12}$$

$$d(t) = 2\mu \int_0^t e(\tau)u(\tau)\, d\tau \tag{13}$$

where all the required gradient signals, $\alpha_{ij}(t)$, $\beta_i(t)$, $x_i(t)$, and $u(t)$ can be obtained by realizing the systems shown in Fig. 4. In Fig. 4, the programmable filter is shown as two separate blocks corresponding to the state-space describing equations. Specifically, the feedback matrix, $\boldsymbol{A}$,

[2] In some cases, it is necessary to use a generalized state-space description to maintain a one-to-one correspondence between the filter's signal-flow-graph coefficients and the state-space elements [14].

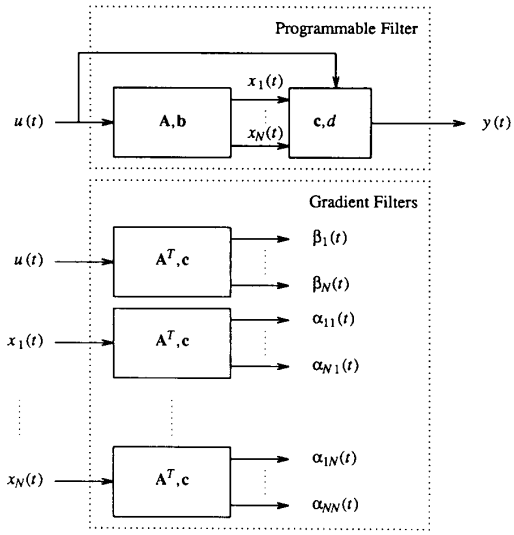


Fig. 4. Generating the gradient signals for a general state-space adaptive filter.

and 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. For the gradient filters, each of the blocks with $\boldsymbol{A}^T$ and $\boldsymbol{c}$ also implement the first equation of the state-space system and are part of the transposed system (also called the adjoint) of the programmable filter. The transposed filter with $u(n)$ as its input is used to obtain the gradients necessary to adapt the $\boldsymbol{b}$ vector while each of the other transposed filters is used to obtain gradients to adapt a single column of the $\boldsymbol{A}$ matrix. The gradients required to adapt the $\boldsymbol{c}$ vector are the states, $x_i(t)$, of the programmable filter.

As described in [1], rather than adapting elements occurring in arbitrary locations in a state-space system, one can significantly reduce the number of computations involved by choosing to adapt the $\boldsymbol{c}$ vector and a single column or row of the $\boldsymbol{A}$ matrix. It was shown that in applications where estimates of final pole locations are known (or equivalently, only the fine-tuning of a transfer function is required), these single column or row filters can significantly improve the dynamic range and adaptation performance over the traditional direct-form structure. It should be mentioned here that these same filters can obtain arbitrary pole and zero locations assuming an observability or controllability condition is met. In the case of a single-column filter, it is clear from Fig. 4 that only one transposed filter is required to obtain the necessary gradient signals. The case of a single-row adaptive filter is shown in Fig. 5 where all the necessary gradient signals are shown to adapt the Nth row and $\boldsymbol{c}$ vector. The $\boldsymbol{b}$ vector contains only one nonzero element in the Nth row and is shown as the basis vector $\boldsymbol{v}_N$. Note that no transposed filters are required since a single row rather than column is being adapted.

IV. The Effects of dc Offsets

For analog adaptive filters, it is well known that dc offsets present in the LMS integrators (shown in Fig. 3(b)) can affect system performance [15]. The dc offsets at these locations cause the filter coefficients to be incorrect resulting in an error in the programmable filter's transfer function at all frequencies (not just at dc). To account for the effects of these offsets, analytical results have been derived for the LMS linear combiner case where only zeros are adjusted [15]–[17]. The purpose of this section is to show that these same formulas can be used to give *approximate* results for the adaptive IIR case where both poles and zeros are adapted. In addition, the analysis of dc offset effects will be extended to account for adaptive filters making use of the *sign-data* LMS algorithm. The sign-data LMS algorithm is a variation on the standard LMS algorithm and is often used in practice due to its low circuit complexity [18].

4.1. dc Offset for the LMS Algorithm

With a dc offset term present at the ith coefficient-update integrator, the LMS update formula (4) above becomes

$$p_i(t) = 2\mu \int_0^t [ke(\tau)\phi_i(\tau) + m_i]\, d\tau \tag{14}$$

where m_i is the dc offset at the ith coefficient-update integrator.

At steady state, the coefficient signal p_i is a constant value implying

$$E[ke(t)\phi_i(t) + m_i] = 0. \tag{15}$$

Since m_i is a dc level, we can write

$$E[e(t)\phi_i(t)] = \frac{-m_i}{k}. \tag{16}$$

Now making the assumption that only small coefficient changes occur due to dc offsets, at a minimum the error signal can be written as

$$e(t) \approx \delta(t) - y^*(t) - \sum_{i=1}^{2N} \frac{\partial y(t)}{\partial p_i} \Delta p_i \tag{17}$$

where $y^*(t)$ is defined as the optimum output that causes the minimum mean-squared error, and Δp_i is defined to be the change in coefficients from their optimum values due to dc offsets. Making use of vector notation, we write the gradient signals, $(\partial y(t)/(\partial p_i)$, as the vector $\boldsymbol{\phi}(t)$ and the change in coefficients, Δp_i, as the vector $\boldsymbol{q}$. Also, since we are interested in finding the *excess* mean-squared error due to dc offsets (as opposed to overall mean-squared error), without loss of generality, we make the assumption that the optimum filter output, $y^*(t)$, equals the reference signal, $\delta(t)$. Therefore, the excess error

signal can be written as

$$e(t) \approx -\phi^T(t)\boldsymbol{q}. \tag{18}$$

Writing the dc offsets as a vector $\boldsymbol{m}$ in (16) and combining it with (18) results in

$$E\left[\phi(t)\phi^T(t)\boldsymbol{q}\right] \approx \frac{\boldsymbol{m}}{k}. \tag{19}$$

Now, defining a gradient correlation matrix, $\boldsymbol{R}$, as

$$\boldsymbol{R} \equiv E\left[\phi(t)\phi^T(t)\right] \tag{20}$$

and solving (19) results in the following formula relating the change in coefficients, $\boldsymbol{q}$, to the dc offsets and the gradient correlation matrix.

$$\boldsymbol{q} \approx \frac{1}{k}\boldsymbol{R}^{-1}\boldsymbol{m}. \tag{21}$$

To obtain the excess mean-squared error, $\|e\|^2$, due to dc offsets, the definition of mean-squared error is used with the following manipulations.

$$\begin{aligned}\|e\|^2 &= E\left[e(t)e(t)\right]\\ &\approx E\left[\boldsymbol{q}^T\phi(t)\phi^T(t)\boldsymbol{q}\right]\\ &\approx \frac{1}{k^2}E\left[\boldsymbol{m}^T\boldsymbol{R}^{-T}\boldsymbol{R}\boldsymbol{R}^{-1}\boldsymbol{m}\right]\\ &\approx \frac{1}{k^2}\boldsymbol{m}^T\boldsymbol{R}^{-1}\boldsymbol{m}.\end{aligned} \tag{22}$$

Note from (22) that the value of the excess error due to dc offsets is proportional to the inverse of the correlation matrix, $\boldsymbol{R}$. This fact implies that the excess error will increase as the matrix $\boldsymbol{R}$ becomes more ill-conditioned as a result of the gradient signals becoming more correlated. This increased excess error is one reason to look for adaptive recursive structures with near orthonormal gradients. Also note that the level of the input signal affects the excess error through $\boldsymbol{R}$. Finally, note that the excess root-mean-squared error value is inversely proportional to the error amplifier gain, k, and therefore, this amplifier can be used to reduce dc offset effects [16]. It should be emphasized that the gain factor, k, has another effect; it increases the adaptation rate, which could cause the adaptive loop to go unstable. For this reason, it is usually necessary to increase the time constant of the LMS integrator by the amount of k. In other words, the step size, μ, should be reduced by the same amount that k is increased.

It should be mentioned here that in the adaptive linear combiner case, the correlation matrix is not a function of the coefficients, p_i, and thus the offset-induced excess error does not depend on the final transfer function of the adaptive filter. However, in the adaptive recursive case, the correlation matrix is a function of the adaptive filter's transfer function, and therefore, one requires a knowledge of the final transfer function to apply the above offset-induced excess error formula. Since this final transfer function is usually not known, one can only hope to obtain approximate results by estimating the correlation matrix assuming some estimate of the final transfer function. This further approximation should not be a major hindrance for design purposes since dc offset estimates themselves are rough approximations.

4.2. dc Offset for the Sign-Data LMS Algorithm

In the sign-data algorithm, the following update equation is used for the coefficient p_i:

$$p_i(t) = 2\mu\int_0^t \left[e(\tau)\,\mathrm{sgn}\left[\phi(\tau)\right]\right] d\tau \tag{23}$$

where sgn[•] represents the signum function. It is not difficult to show that when dc offsets are present in a system using the sign-data algorithm, the following equations result for the change in coefficients and excess mean-squared error.

$$\boldsymbol{q} \approx \frac{1}{k}\tilde{\boldsymbol{R}}^{-1}\boldsymbol{m} \tag{24}$$

and

$$\|e\|^2 \approx \frac{1}{k^2}\boldsymbol{m}^T\tilde{\boldsymbol{R}}^{-T}\boldsymbol{R}\tilde{\boldsymbol{R}}^{-1}\boldsymbol{m} \tag{25}$$

where $\boldsymbol{R}$ is defined as before and the elements of the signum correlation matrix, $\tilde{\boldsymbol{R}}$ are defined as

$$\tilde{R}_{ij} = E\left[\mathrm{sgn}\left[\phi_i(t)\right]\phi_j(t)\right]. \tag{26}$$

To apply these formulas for the case of white-noise inputs, the matrix $\boldsymbol{R}$ can be easily obtained using impulse responses; however, it is not clear how one can easily obtain the matrix $\tilde{\boldsymbol{R}}$. Fortunately, in the special case where the input has a *Gaussian* white-noise zero-mean characteristic, one can find a closed-form expression for the elements of $\tilde{\boldsymbol{R}}$ in terms of $\boldsymbol{R}$. Specifically, if the input signal has a zero-mean Gaussian distribution, the joint probability density function, $\Phi_{x_ix_j}(x_i, x_j)$, between the signals x_i and x_j can be written as [19].

$$\Phi_{x_ix_j}(x_i, x_j) = \left(\frac{1}{4\pi^2\left[R_{ii}R_{jj} - R_{ij}^2\right]}\right)^{1/2} \cdot \exp\left(\frac{-1}{2}\left[x_1 x_2\right]\begin{bmatrix} R_{ii} & R_{ij}\\ R_{ij} & R_{jj}\end{bmatrix}^{-1}\begin{bmatrix} x_1\\ x_2\end{bmatrix}\right). \tag{27}$$

Using this joint probability function, the term $\tilde{R}_{ij}$ can be found by integrating the weighted probability density function over both variables:

$$\tilde{R}_{ij} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{sgn}\left[x_1\right]x_2\Phi_{x_ix_j}(x_i, x_j)\,dx_1\,dx_2. \tag{28}$$

Performing this integration leads to the following closed-

form expression[3] for the elements of $\tilde{\boldsymbol{R}}$.

$$\tilde{R}_{ij} = \left(\frac{2R_{ij}^2}{\pi R_{ii}} \right)^{1/2}. \tag{29}$$

In summary, for the case of zero-mean Gaussian white-noise inputs, (29) can be used to obtain $\tilde{\boldsymbol{R}}$ and (25) can be used to find the approximate offset-induced excess error for adaptive IIR filters using the sign-data algorithm. Note that this same formula gives exact results for the adaptive linear combiner case using the sign-data algorithm.

4.3. dc Offset Example

In this section, the advantages of choosing a good filter structure for dc offset effects will be demonstrated by an example. The performance measure used to compare the dc offset effect for different filter structures is denoted as Γ_{off} and is defined as follows.

$$\Gamma_{\text{off}} = \sum_{i=1}^{2N} \boldsymbol{v}_i^T \tilde{\boldsymbol{R}}^{-T} \boldsymbol{R} \tilde{\boldsymbol{R}}^{-1} \boldsymbol{v}_i. \tag{30}$$

Here, $2N$ is the number of adjustable filter coefficients and $\boldsymbol{v}_i$ is a basis vector where the unity element is in the ith row. Qualitatively, this performance measure, Γ_{off}, gives the value of the excess mean-squared error averaged over all the different combinations of sign values for coefficient-update integrator dc offsets having a magnitude value of one.

For the dc offset example, consider a model-matching application where the normalized reference filter is a fourth-order bandpass filter with a pair of zeros at each of dc and ∞ and poles at $-0.05331 \pm j1.0106$ and $-0.04684 \pm j0.8880$ (each pole-Q is about 10). The companion form filter which realizes this transfer function can be described by the following state-space coefficients.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.81 & -0.18027 & -1.82506 & -0.2003 \end{bmatrix},$$

$$\boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.1989 \end{bmatrix}.$$

$$\boldsymbol{c}^T = [0 \quad 0 \quad 1.2346 \quad 0], \quad d = 0. \tag{31}$$

Using the above system in an adaptive filter application would result in eight variable coefficients consisting of the bottom row of $\boldsymbol{A}$ matrix and the entire $\boldsymbol{c}$ vector. Calculating the necessary 8×8 $\boldsymbol{R}$ and $\tilde{\boldsymbol{R}}$ matrices and applying (30) results in the value of the performance measure, Γ_{off}, to be 3585.

[3]It was pointed out by a reviewer that the same result can more easily be obtained through the use of Price's theorem [20] of which a version more suited to our present needs is given in [21].

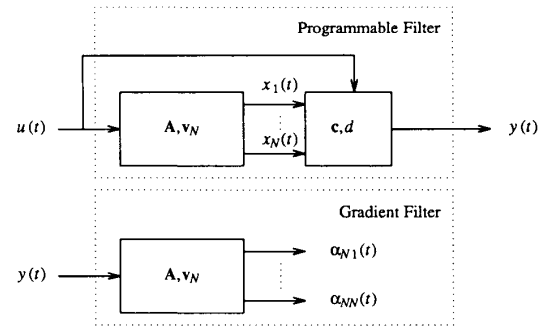


Fig. 5. Generating the gradient signals for a single-row adaptive filter.

Now consider realizing the same transfer function using an orthonormal ladder structure [22]. Such an adaptive filter implementation would result in the following state-space coefficients.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0.9357 & 0 & 0 \\ -0.9357 & 0 & 0.1561 & 0 \\ 0 & -0.1561 & 0 & 0.9618 \\ 0 & 0 & -0.9618 & -0.2003 \end{bmatrix},$$

$$\boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.2525 \end{bmatrix}.$$

$$\boldsymbol{c}^T = [-0.606 \quad 0 \quad 0.101 \quad 0], \quad d = 0. \tag{32}$$

For this structure, the variable coefficients would be the entire $\boldsymbol{c}$ vector and each of the $\boldsymbol{A}$ matrix elements with the constraint that the shown symmetry is maintained. The performance measure, Γ_{off}, for this orthonormal structure is calculated to be only 66! This number implies that one could allow approximately seven times the amount of dc offset in the orthonormal design as compared to the companion form design and yet achieve the same dc offset performance. Although these numbers are application specific, it is clear that the choice of filter structure significantly affects the dc offset performance.

V. Discrete Prototype and Experimental Results

To verify many of the theoretical derivations and check the practicality of this analog adaptive filter approach, a third-order discrete prototype was constructed and tested. The prototype is based on the single-row form shown in Fig. 5 and, for evaluation purposes, the poles of the programmable filter are placed at low frequencies—in the 1-KHz range. The basic structure of the feedback systems used in both the programmable and gradient filters is the orthonormal ladder structure [22]. For further design details of the discrete prototype, the reader is referred to [23].

The block diagram of the prototype is shown in Fig. 6. There are six coefficients used to adjust the transfer function of the programmable filter: A_{3i}, $i = 1-3$ and c_i,

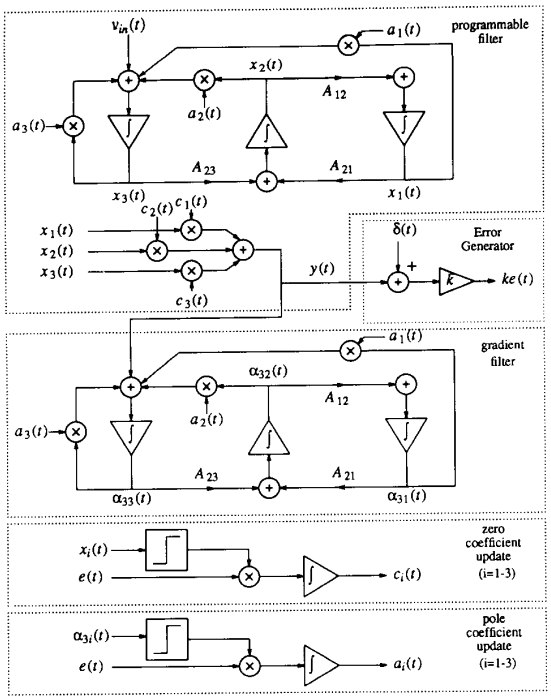


Fig. 6. Block diagram of a third-order single-row analog adaptive filter.

$i = 1-3$. As shown, three pole coefficient update blocks are used to adapt the A_{3i} coefficients while the c_i coefficients are adapted using three zero coefficient update blocks. Note that these coefficient update blocks use the sign-data LMS algorithm to simplify the multiplication between the gradient and error signals. The gradient signals needed to adjust the c_i coefficients are simply the states, $x_i(t)$, of the programmable filter whereas to adjust the a_i coefficients, the gradient signals, $\alpha_{3i}(t)$, are obtained as the outputs from a gradient filter. Note that the gradient filter is identical to the feedback circuit used in the programmable filter.

Referring to Fig. 6, we see that the sign of the gradient signal is multiplied by $ke(t)$ where k is the amplification constant applied to the error signal to reduce the offset effects as discussed above. Experimentation confirmed the reduction in offset-induced excess error when increasing the gain factor, k. It should be pointed out that this gain factor was arbitrarily chosen to be 82 for the discrete prototype and will be difficult to realize for high-frequency circuits. This difficulty in implementation is one of the major reasons for developing the dc offset formulas. With these formulas available and a known tolerance on dc offsets, a designer can choose the minimum error gain factor, k, necessary to meet specifications.

Referring again to Fig. 6, it is clear that multiplier/summer circuits are required for the programmable and gradient filters. The circuits realizing these multiplier/summer stages are based on a MOSFET linearization technique originally proposed for creating fixed continuous-time integrated filters [24]. Of course, as discussed earlier, many different techniques could be used to obtain these variable filter coefficients in an integrated version.

Since single-row adaptive filters require some estimate of final pole locations, it was decided to choose component values so that the nonadapting coefficients corresponded to those used in the reference filter of the first experiment discussed below. Thus the following normalized state-space filter was implemented where the coefficients to be adapted are shown as variables.

$$A = \begin{bmatrix} 0 & 0.98361 & 0 \\ -0.98361 & 0 & 1.2307 \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ 0.7737 \end{bmatrix}$$
$$c^T = [c_1 \quad c_2 \quad c_3], \quad d = 0. \tag{33}$$

This normalized state-space system and others to be described were all denormalized such that time-constant values were placed around the 1-KHz range.

5.1. Model-Matching Experimental Results

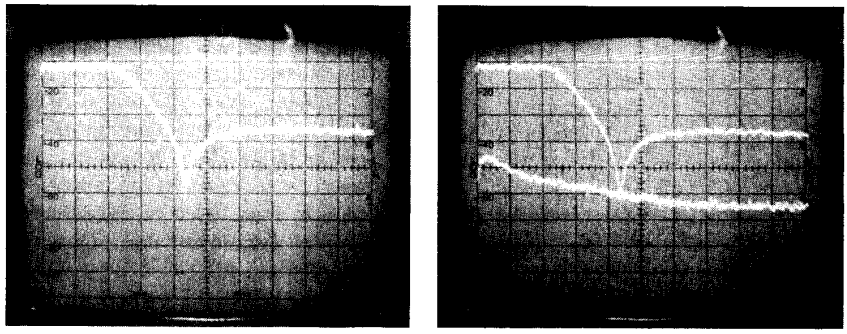
In order to test the adaptive filter, the model-matching application was employed where a white-noise source was applied at both the inputs of the adaptive and reference filters. In the first experimental example, the reference filter was a third-order low-pass filter with finite transmission zeros and a pair of complex poles with natural frequency and pole-Q equal to 1.3 and 3.3, respectively. The normalized state-space system for the reference filter was

$$A = \begin{bmatrix} 0 & 0.98361 & 0 \\ -0.98361 & 0 & 1.2307 \\ 0 & -1.2307 & -1.8805 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ 0.7737 \end{bmatrix}$$
$$c^T = [1.5779 \quad 0 \quad 0.4563], \quad d = 0. \tag{34}$$

Note that except for the coefficients that adapt, this system is the same as that in (33) and therefore, after adaptation, the coefficients A_{31}, A_{32}, and A_{33} should correspond to 0, -1.2307, and -1.8805, respectively, while c_1, c_2, and c_3 should correspond to 1.5779, 0, and 0.4563, respectively. Thus this example corresponds to the case where a good structure (the orthonormal structure) has been chosen and one knows the exact location of final poles. Although this is not a realistic case, it is the first experimental result presented.

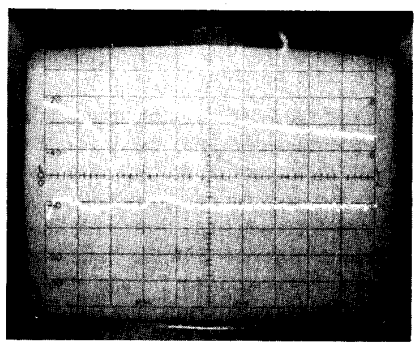
Fig. 7(a) shows the close matching of the spectra of the reference and adaptive filters after approximately 1 s of adaptation. To determine the level of mismatch between the two spectra, the spectra of the error and reference signal are plotted together in Fig. 7(b). Note that the error signal is approximately 40 dB below the level of the reference signal indicating a close level of matching.

In a more realistic experiment, the circuitry for the adaptive filter was left unchanged and adaptation was performed to two different reference filters as shown in Fig. 8. In Fig. 8(a), the reference filter was a third-order notch filter with the complex poles having a natural

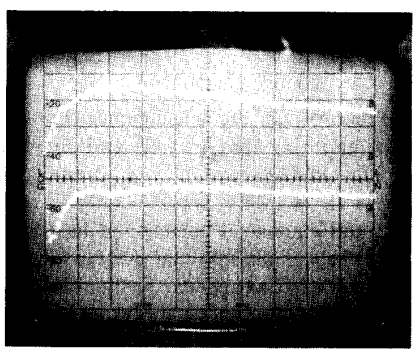


(a) (b)

Fig. 7. Experimental results for third-order low-pass reference filter. Vertical scale = 10 dB/div, Horizontal scale = 500 Hz/div. (a) Signal spectra for $\delta(t)$ and $y(t)$. (b) Signal spectra for $\delta(t)$ and $e(t)$.



(a)



(b)

Fig. 8. Experimental results for notch and bandpass examples. In both cases, signal spectra for $\delta(t)$ and $e(t)$ are shown. Vertical scale = 10 dB/div, Horizontal scale = 500 Hz/div. (a) Third-order notch reference filter. (b) Second-order bandpass reference filter.

frequency and pole-Q of 1.6387 and 4.85, respectively. These poles are significantly different than the previous example, yet as seen in Fig. 8(a), the adaptive filter successfully matched the reference filter. Finally, as shown in Fig. 8(b), the third-order reference filter was modified to a second-order bandpass filter (with a pole-Q of 0.7) implying the adaptive filter must match a lower order system. For this example, a zero and pole become coincident causing the adaptive filter's transfer function to be reduced from third to second order. With this type of cancellation, one could argue that the cancelled pole-zero pair might move about in the s-plane and possibly go into the unstable region of the plane. However, this was not observed during experimentation although the setup was left running for well over an hour.

5.2. dc Offset Experimental Results

In this section, predicted excess mean-squared error due to dc offsets is compared with experimental results. As the discrete prototype makes use of the sign-data algorithm, the sign-data excess error formula (25) will be used for predicting the offset effects.

Before proceeding with the comparison, note that (25) predicts the excess rms error that would result from a known dc offset applied to an initially offset-free circuit. However, unknown dc offsets always exist in the discrete prototype and therefore must be taken into account as follows. First, measure the rms error, $\|ke\|_{m_1}$, due to the unknown dc offsets, $\boldsymbol{m}_1$, always present in the circuit. Next, apply a known set of dc offsets, $\boldsymbol{m}_2$, to the circuit and measure the rms error, $\|ke\|_p$, due to $\boldsymbol{m}_1 + \boldsymbol{m}_2$. Then apply the opposite polarity of known offsets, $-\boldsymbol{m}_2$, to measure $\|ke\|_n$ due to offsets consisting of $\boldsymbol{m}_1 - \boldsymbol{m}_2$. Finally, the following relationship can easily be derived to determine the rms error, $\|ke\|_{m_2}$, due to a set of $\boldsymbol{m}_2$ offsets alone.

$$\|ke\|_{m_2}^2 = 1/2\|ke\|_p^2 + 1/2\|ke\|_n^2 - \|ke\|_{m_1}^2. \tag{35}$$

Using the above approach, experimental versus theoretical results for dc offsets are compared in Table I. For these results, the reference and adaptive filters correspond to (33) and (34) respectively, and each row of Table I corresponds to a known dc offset vector of $\pm 0.122\boldsymbol{v}_i$

TABLE I
A COMPARISON OF THEORETICAL AND EXPERIMENTAL RESULTS FOR INJECTED dc OFFSETS. WHEN NO dc OFFSETS APPLIED, $\|ke\|_{m_1}$ EQUALS 0.2 Vrms. THE iTH ROW CORRESPONDS TO dc OFFSETS ON THE iTH INTEGRATOR

i	Corresponding Coefficient	$\|ke\|_p$	$\|ke\|_n$	Experimental $\|ke\|_{m_2}$	Theoretical $\|ke\|_{m_2}$	Percentage Error
1	c_1	1.1	0.7	0.9	0.77	17
2	c_2	0.31	0.36	0.27	0.281	−4
3	c_3	0.33	0.26	0.22	0.22	0
4	A_{31}	0.4	0.34	0.31	0.275	13
5	A_{32}	0.7	0.51	0.58	0.575	1
6	A_{33}	0.7	0.46	0.56	0.538	4

where v_i is a basis vector with a value of unity in the ith row.[4]

To measure the rms voltages of the amplified error, $\|ke\|$, a digital-readout true rms meter was used where it was not a simple matter to obtain the rms value of the output error signal due to the presence of low-frequency signal components. However, we see from Table I that all measurements agree within 20% of the theoretical predictions, a reasonable degree of accuracy considering that all circuit nonidealities other than integrator dc offsets have been ignored and that rms measurements involving noise signals are used.

VI. CONCLUSIONS

A design approach for continuous-time LMS adaptive recursive filters was presented. These analog adaptive filters should be capable of higher frequency operation than their digital counterparts. However, nonideal effects such as dc offsets exist that are not present in digital realizations and so formulas were developed to predict the effects of these dc offsets for both the LMS and the sign-data LMS algorithms. As well, it was shown that the choice of filter structure and amplification of the error signal can reduce the dc offset effects. Finally, experimental results with a discrete prototype verified operation of the adaptive technique and the dc offset formulas derived.

ACKNOWLEDGMENT

The authors would like to thank F. Kschischang for aid in developing the sign-data dc offset formulas.

REFERENCES

[1] 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, pp. 673–684, June 1990.
[2] J. J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Magazine*, vol. 6, pp. 4–21, Apr. 1989.
[3] R. Schaumann, M. S. Ghausi, and K. R. Laker, *Design of Analog Filters: Passive, Active RC and Switched Capacitor*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
[4] K. A. Kozma, D. A. Johns, and A. S. Sedra, "An adaptive tuning circuit for integrated continuous-time filters," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1163–1166, May 1990.
[5] G. A. Bekey and R. B. McGhee, "Gradient methods for the optimization of dynamic system parameters by hybrid computation," in *Computing Methods in Optimization Problems*, A. V. Balakrishnan and L. W. Neustadt, Eds. New York: Academic Press, pp. 305–327, 1964.
[6] G. A. Bekey and W. J. Karplus, *Hybrid Computation*. New York: Wiley, 1968.
[7] W. B. Mikhael and F. F. Yassa, "An implementation of a continuous-time frequency-domain adaptive system," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 585–588, May 1982.
[8] ____, "Stable high order continuous adaptive filters," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 666–669, May 1982.
[9] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[10] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143–2159, Dec. 1967.
[11] W. M. Snelgrove and A. S. Sedra, "Synthesis and analysis of state-space active filters using intermediate transfer functions," *IEEE Trans. Circuits Syst.*, vol. 33, pp. 287–301, Mar. 1986.
[12] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[13] K. W. Martin and M. T. Sun, "Adaptive filters suitable for real-time spectral analysis," *IEEE Trans. Circuits Syst.*, vol. 33, pp. 218–229, Feb. 1986.
[14] G. W. Roberts and A. S. Sedra, "A generalization of intermediate transfer function analysis applied to arbitrary networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1059–1062, May 1989.
[15] R. T. Compton, Jr., *Adaptive Antennas*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[16] K. R. Perry, "A distributed-μ implementation of the LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, pp. 753–762, June 1981.
[17] C.-P. J. Tzeng, "An adaptive offset cancellation technique for adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 799–803, May 1990.
[18] J. R. Treichler, C. R. Johnson, and M. G. Larimore, *Theory and Design of Adaptive Filters*. New York: Wiley, 1987.
[19] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, p. 186, 1984.
[20] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IEEE Trans. Inform. Theory*, vol. IT-4, pp. 69–72, June 1958.
[21] N. J. Bershad, "On error-saturation nonlinearities in LMS adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 440–452, Apr. 1988.
[22] D. A. Johns, W. M. Snelgrove, and A. S. Sedra, "Orthonormal ladder filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 337–343, Mar. 1989.
[23] ____, "Continuous-time analog adaptive recursive filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 667–670, May 1989.
[24] M. Banu and T. Tsividis, "Fully integrated active *RC* filters in MOS technology," *IEEE J. Solid-State Circuits*, vol. 18, pp. 664–651, Dec. 1983.

[4]The value of 0.122 corresponds to a resistor choice used to inject a current into the virtual-ground of the LMS integrator resulting in an equivalent dc offset of 122 mV.

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

From 1980 to 1981 he worked as an applications engineer in the semiconductor division of Mitel Corp., 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 was hired at 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 INAOE 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. In 1989 and 1990 he spent six-month sabbatical leaves as a resident visitor at AT&T Bell Labs in Reading, PA, working in CMOS analog design. He is active in integrated and adaptive signal-processing systems, CAD for filter design, and multiprocessors for signal processing and CAD. 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.