Automatic Tuning of Continuous-Time Integrated Filters Using an Adaptive Filter Technique

Karen A. Kozma, David A. Johns, *Member, IEEE*, and Adel S. Sedra, *Fellow, IEEE*

Abstract—**An adaptive filter technique for tuning continuous-time integrated filters is presented. This adaptive technique is based on the model-matching configuration and tunes both the poles and zeros of the transfer function. Circuit details of an experimental prototype are given. The experimental prototype consists of an integrated third-order filter that is automatically tuned by off-chip circuitry realizing the adaptive tuning system. Both experimental and simulation results are presented to confirm the viability of the proposed approach.**

I. Introduction

WITHOUT tuning, the transfer function of a continuous-time integrated filter will vary considerably due to fabrication tolerances, environmental changes, parasitic effects, and the finite input and output impedances of the integrators [1], [2]. In order to reduce this transfer-function variability, automatic tuning circuits have been developed. The present approach for automatic tuning (known in the literature as the master/slave tuning method) usually adjusts the pole frequencies and quality factors of a model while relying upon matching between the corresponding elements of the model and the filter to be tuned [1]–[9]. Exactly how the critical frequencies and quality factors are tuned depends upon the technology used. For example, in MOSFET-C technology, an equivalent voltage-controlled resistor can be obtained by adjusting the gate voltage of a transistor (operated in the triode region) [10], whereas, in transconductance-C technology, a variable coefficient can be obtained by altering the bias current of transconductance cells [1]–[9]. Because present schemes rely upon element matching, both the model and the filter should be physically near each other. Unfortunately, being close together encourages the feedthrough of control signals, originating in the tuning circuitry, to the filter. Thus the physical distance between the filter and the model should be a compromise between good matching properties and low noise interference. As well, since the present schemes tune the model rather than the filter, the transfer function of the filter itself is never evaluated.

The tuning scheme proposed in this paper is based on the model-matching configuration, which is well known to adaptive filter designers. With this adaptive tuning approach, the tuned transfer-function is continually monitored since the output of the filter is used to operate the tuning circuitry. There are no critical matching dependencies between integrated filter elements thus allowing the tuning circuit and filter to be well isolated from each other. Furthermore, whereas present approaches only tune the poles, the adaptive tuning scheme tunes all the poles and *zeros*, thereby enabling a closer match to the desired transfer function. However, it is expected that the proposed scheme will use slightly more silicon area than a master/slave tuning system where each pole-pair is individually tuned. As well, this adaptive tuning scheme requires two low-resolution digital-to-analog converters (DAC's) operating at a frequency greater than the passband of the tunable filter.

An overview of this paper is as follows. The adaptive tuning approach is introduced in Section II, and many of the details on the implementation of the approach are discussed in Sections III and IV. Simulation results are given to show the validity of the proposed tuning method in Section V. Finally, circuit details and experimental results of a prototype, realizing the adaptive tuning system, are presented in Sections VI and VII, respectively. The prototype consists of an integrated CMOS filter and off-chip tuning circuitry.

II. The Adaptive Tuning Approach

As mentioned above, the adaptive tuning approach is based on the model-matching system shown in Fig. 1(a). The basic concept underlying the model-matching configuration is to adjust the transfer function of a tunable filter to be the same as that of a reference filter. This goal is accomplished by applying a spectrally rich signal (usually white noise) to the inputs of the tunable and reference filters and then utilizing an adaptive algorithm to adjust the coefficients of the tunable filter so as to minimize some measure of an error signal, $e(t)$. The error signal,

Manuscript received February 4, 1991; revised July 8, 1991. This paper was recommended by Associate Editor D. J. Allstot.

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

IEEE Log Number 9103419.

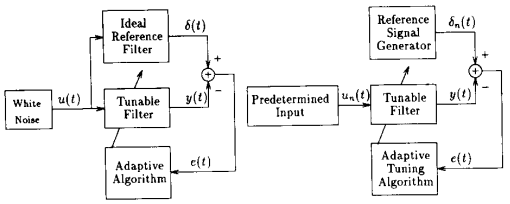


Fig. 1. Model-matching configuration. (a) Traditional configuration. (b) Adaptive tuning system (modified configuration by replacing the ideal reference filter).

$e(t)$, is simply the difference between the output of the reference filter, $\delta(t)$, and the output of the tunable filter, $y(t)$.

The model-matching configuration in Fig. 1(a) suggests that, given an ideal reference filter, the tunable filter may be tuned to a desired transfer function. However, the assumption that an "ideal" filter already exists obviously conflicts with the fundamental need for tuning. The adaptive tuning system, shown in Fig. 1(b), eliminates the need for the ideal filter as follows. The white noise input is replaced by a predetermined input, $u_n(t)$, so that the corresponding output, $\delta(t)$, can be precalculated. For implementation simplicity, the precalculated output, $\delta(t)$, is replaced by a discrete-time version, $\delta_n(t)$. We will show in the next section that the input, $u_n(t)$, and the reference signal, $\delta_n(t)$, can be efficiently realized in terms of silicon area.

Although the adaptive tuning system in Fig. 1(b) suggests the filter cannot service the system input during tuning, this need not be the case. For example, two tunable filters may be included on the same chip so that one services the system input while the other is being tuned by the adaptive loop [11]. Switching between the filter in service and the filter being tuned will then facilitate the tuning of both filters as well as servicing the input. An alternative method that does not involve interrupting the filter that produces the system output may also be implemented [12], [13]. Once again two filters are used where one filter (A) continuously services the input. The second filter (B) is first tuned by the adaptive loop so that it may be used as a "reference filter" for tuning filter A. Once filter B is tuned, the actual input signal is applied to filter B creating a reference signal, $\delta(t)$, used to tune filter A. Thus this method assumes the spectral content of the system input is sufficiently rich to fully characterize the desired transfer function. Note, however, that no critical matching is required between the integrated elements of filters A and B, and the system output is never interrupted.

Before proceeding, a few comments are worth making. First, note that the type of adaptive algorithm has not been specified. In fact, any adaptive algorithm that performs well in the model-matching configuration can be used. For practical reasons, we have chosen to use the least-mean-squared (LMS) algorithm[1] which minimizes the mean-squared-error (MSE) value [14]. Thus, for the remainder of this paper, we assume the use of the LMS algorithm. Also note that no mention has been made of the structure of the tunable filter. Ideally, any structure can be used with the constraint that the variable elements should allow placement of the poles and zeros to their desired locations. However, it has been shown that the choice of filter structure can significantly affect the overall performance of an adaptive system [15]. Also, note the feedback in this adaptive tuning approach: The coefficients of the tunable filter are adjusted through the use of the adaptive algorithm, and the output, $y(t)$, is continually compared to the reference signal, $\delta_n(t)$, to minimize the MSE value. Thus the accuracy of the final transfer function is dependent on the accuracies of the adaptive algorithm, the input signal, and the reference signal, rather than on the matching between a model and the filter. Finally, since the method makes use of an IIR LMS algorithm in a model-matching configuration, one might have to be concerned with converging to a local minimum. Fortunately, theoretical results have been derived proving that as long as the order of the tunable filter is at least as high as that of the reference filter, and that all poles and zeros are adjusted, then local minima do not exist [16]. Thus, for the specific case of tuning filters, converging to a local minimum is not a problem since the order of the tunable filter can be chosen to be the same as that of the known reference filter.

III. Generation of Suitable Input and Reference Signals

For the adaptive tuning system in Fig. 1(b) to perform as well as the traditional model-matching scheme, the predetermined input, $u_n(t)$, should be spectrally rich enough to fully characterize the filter being tuned, and the reference signal, $\delta_n(t)$, should suitably replace the function of the ideal reference filter. As well, for practical purposes, the input and reference signals should be generated by circuitry that is simple yet accurate. Appropriate input and reference signals are presented here.

For a spectrally rich input signal, we propose the use of a pseudorandom noise (PN) sequence. It is well known that PN sequences can be generated through the use of shift registers and some simple logic, with the length of the PN sequence determined by $length = 2^{\#registers} - 1$ [17]. Perhaps not as well known is the fact that the length of the PN sequence influences how closely the sequence resembles white noise. Specifically, for a fixed repetition rate, an increase in the length of the PN sequence results in the sequence more closely resembling white noise for frequencies above the repetition rate. The next section will present simulation results demonstrating how the spectral richness of the input signal can affect tuning performance.

[1]The LMS algorithm will be defined in Section III.

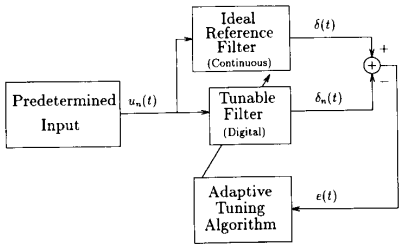


Fig. 2. Simulation setup for calculating the discrete-time reference signal, $\delta_n(t)$.

To generate the reference signal, we make use of the fact that at steady-state, a periodic input fed to a linear system results in a periodic output at the same repetition rate. Therefore, since the proposed input is periodic and the ideal continuous-time reference filter is linear, it is clear that the reference signal must also be a continuous-time periodic signal. However, since it is difficult to generate arbitrary continuous-time signals, we propose the use of discrete levels for the reference signal. As will be seen, exact tuning behavior can be obtained by using a relatively small number of discrete levels for both the input signal and the reference signal. Consequently, both signals can be generated by making use of area efficient circuits.

We next address the issue of determining a set of values that optimally represent the reference signal. Referring to the adaptive tuning system in Fig. 1(b), it is evident that the adaptive algorithm adjusts the coefficients of the tunable filter until the MSE value is minimized. A key point here is that an adaptive algorithm is used to vary the continuous-time signal, $y(t)$, to best match the discrete-time signal, $\delta_n(t)$, by minimizing the MSE value. An obvious approach to obtain the discrete levels, $\delta_n(t)$, would be to uniformly sample the desired continuous-time signal. However, it was found that an extremely high number of samples would be necessary to reduce the MSE value between $\delta_n(t)$ and the desired output, $\delta(t)$, to an acceptable level [13]. Thus another approach was employed to obtain a reduced number of discrete levels. Specifically, the discrete-time signal, generated by an Nth-order filter, which best minimizes the MSE value between the discrete levels and the desired continuous-time signal was calculated. A logical procedure to do this would be to simulate the reverse setup of the adaptive tuning system, as shown in Fig. 2. Using the predetermined input, $u_n(t)$, an ideal reference filter produces the desired continuous output, $\delta(t)$, while a digital filter is adapted to produce the reference samples, $\delta_n(t)$. Since the simulation scheme shown in Fig. 2 minimizes the MSE value between $\delta_n(t)$ and $\delta(t)$, upon implementation of the adaptive tuning system, shown in Fig. 1(b), $y(t)$ should be tuned to equal $\delta(t)$.

In terms of circuit implementation, if the number of discrete levels for both the input and reference signals are not large, the system shown in Fig. 3 can be used to

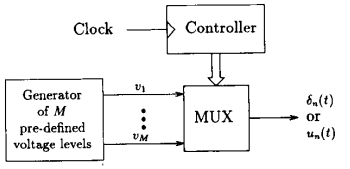


Fig. 3. Block diagram showing the generation of M discrete levels.

generate each. Having a pre-defined set of M voltage levels, a discrete-time signal is generated by clocking out one of the M possible voltage levels at the appropriate time. Note that this implementation is equivalent to using a DAC and selecting one of the M possible output values at the appropriate time. It should be emphasized here that with ideal adaptive circuitry, the transfer-function matching is determined by the accuracy of the levels rather than the number of samples, M, used. Specifically, using 4-bit DAC's ($M = 16$) accurate to 10 bits would give a coefficient accuracy of around 10 bits rather than only 4 bits.

IV. The Effect of the Length of the PN Sequence on the Speed of Tuning

In this section, a simulation example is presented to illustrate the trade-offs between the speed of tuning and the length of, and hence the size of circuitry required to generate, the PN sequence input. For simplicity, the model-matching configuration in Fig. 1(a) was simulated. Specifically, an ideal continuous-time reference filter created the desired output, and the adjoint of the orthonormal ladder filter structure, described in [18], was used for the tunable filter. Both continuous-time filters were simulated by applying the bilinear transform to obtain digital filter equivalents that were clocked at 100 times the respective passband frequencies. As well, PN sequences were substituted for the white noise input. The repetition rates of all the PN sequences corresponded to two-thirds the passband frequency of the desired transfer function. This repetition rate was chosen as an arbitrary compromise between too low a rate, which would interfere with the low adaptation rates of the overall system, and a rate in the stopband of the filter, where the power of the output signals, $\delta_n(t)$, $y(t)$, and $e(t)$, would be too small to enable any useful adaptation.

The reference filter had a third-order low-pass transfer function with the following characteristics:

$$\begin{aligned} \text{poles} &= \{-0.43 \pm j1.25, -0.99\} \\ \text{zeros} &= \{\pm j2.33, \infty\} \\ \text{dc gain} &= 1.46. \end{aligned} \tag{1}$$

A plot of the transfer function of this filter is shown in Fig. 4. As a starting point for the programmable filter, all the feed-forward coefficients (refer to the orthonormal structure in [18]) were set to zero while the feedback coefficients were set such that the following poles were

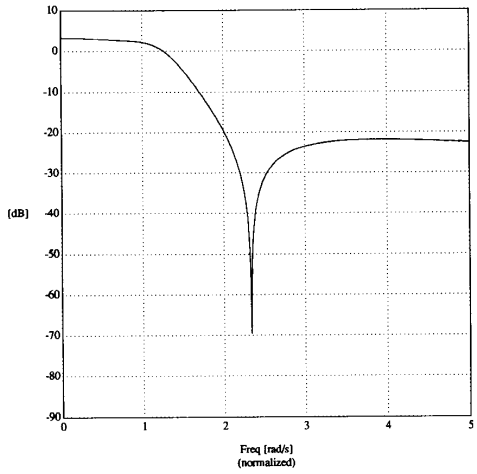


Fig. 4. Plot of the ideal transfer function.

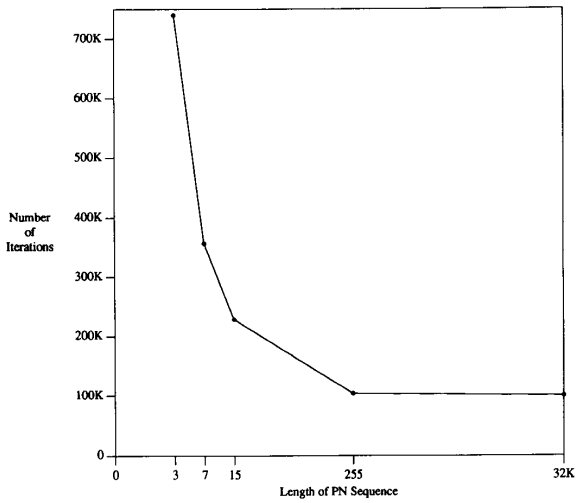


Fig. 5. Number of iterations performed to tune a third-order filter for PN sequences of different lengths.

realized:

$$\text{poles} = \{-0.24 \pm j2.77, -0.52\}. \quad (2)$$

For comparison purposes, the filter was considered tuned when the MSE value was reduced by 75 dB. The number of iterations performed to tune the filter for the different PN inputs are plotted in Fig. 5. As expected, the number of iterations required to tune the filter decreases as the length of the PN sequence is increased. Although only adaptation speed is compared, this measure is usually related to the conditioning of the performance surface. Specifically, a relatively long adaptation time can indicate an ill-conditioned performance surface and hence a poor overall adaptation performance in terms of coefficient matching. Thus we see from this third-order example that a trade-off must be made between the length of the PN sequence and adaptation performance. For higher order filters, the tuning-input signal should more closely resemble white noise and correspondingly, the length of the PN sequence will increase.

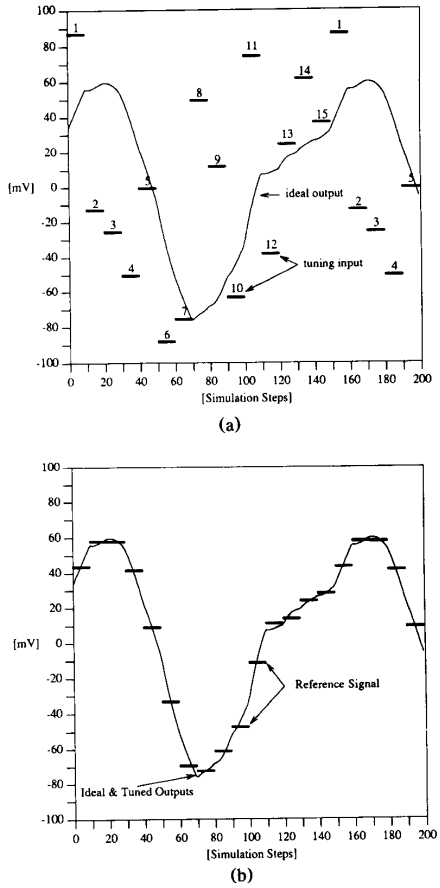


Fig. 6. Plots illustrating the 4-bit PN sequence input, discrete-time reference signal, ideal output, and tuned output. (a) PN sequence input, $u_n(t)$, and ideal output, $\delta(t)$. (b) Discrete-time reference signal, $\delta_n t$, ideal output, $\delta(t)$, and tuned output, $y(t)$.

V. Verification of the Adaptive Tuning System

Another simulation example is presented here to verify the validity of the adaptive tuning system in Fig. 1(b). Once again, the adjoint of the orthonormal ladder filter structure was used for the tunable filter and simulated by the method described in the preceding section. The desired transfer function was the same third-order filter illustrated in Fig. 4, and a four-bit PN input was employed. The 15 horizontal lines in Fig. 6(a) represent the PN samples and the solid curve is the ideal response, $\delta(t)$. Fifteen samples were also chosen to implement the reference signal and were determined by simulating the system shown in Fig. 2. Here, the output of the digital

filter was sampled at the same rate as the PN sequence so that the same clock can control the tuning and reference sequences in a circuit realization of the adaptive tuning system.

After tuning, the filter had the following characteristics:

$$\text{poles} = \{-0.43 \pm j1.24, -1.00\}$$
$$\text{zeros} = \{-0.02 \pm j2.34, \infty\}$$
$$\text{dc gain} = 1.46. \tag{3}$$

Fig. 6(b) illustrates the matching between the desired and tuned outputs. The horizontal lines depict the 15 reference samples while the two solid coincident curves represent the output to which the filter was tuned, $y(t)$, and the ideal output, $\delta(t)$. This close matching between the ideal and tuned outputs was obtained for arbitrary initial starting points for the tunable filter, indicating the absence of local minima.

In summary, the ideal reference filter in the model-matching configuration can be replaced by a PN sequence and reference samples. It was shown that a third-order filter may be tuned to arbitrary precision by using two signals, each consisting of 15 discrete voltage levels.

VI. Design of an Experimental Prototype

For experimental verification, a discrete prototype circuit was constructed to tune a third-order integrated CMOS filter. This section will present some of the circuit details of this prototype system.

For the integrated tunable filter, the adjoint of the orthonormal ladder structure [18] was used which is best described with state-space equations:

$$s\boldsymbol{X}(s) = \boldsymbol{A}\boldsymbol{X}(s) + \boldsymbol{b}U(s)$$
$$Y(s) = \boldsymbol{c}^T\boldsymbol{X}(s) + dU(s) \tag{4}$$

where $U(s)$ is the input signal; $\boldsymbol{X}(s)$ is a vector of N states (for a filter of order N), which in fact are 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 this system can be easily shown to be

$$T(s) = \frac{Y(s)}{U(s)} = \boldsymbol{c}^T(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} + d. \tag{5}$$

The poles are determined by the $\boldsymbol{A}$ coefficients while the coefficients of all four matrices determine the zeros of the transfer function.

For the adjoint of a third-order orthonormal ladder filter, the coefficients, $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and d take the form:

$$\boldsymbol{A} = \begin{bmatrix} 0 & -a_1 & 0 \\ a_1 & 0 & -a_2 \\ 0 & a_2 & -a_3 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$
$$\boldsymbol{c}^T = [0 \quad 0 \quad 1] \qquad d = 0 \tag{6}$$

where d has been set to zero since we will be tuning the filter to a transfer function that has at least one zero at infinity. With this filter structure, it is not difficult to show that arbitrary poles and zeros (with at least one zero at infinity) can be obtained by adjusting the $\boldsymbol{A}$ and $\boldsymbol{b}$ coefficients. The structure also has a simple stability check system [19]:

$$a_3 > 0 \quad \text{and} \quad a_1, a_2 \neq 0. \tag{7}$$

As well, note that the $\boldsymbol{c}$ vector is simply a basis vector and therefore eliminates the need for constructing a separate output summing stage.

A block diagram for this filter structure is shown in Fig. 7. As discussed, the states of the system are defined as the outputs of the integrators, $x_i(t)$, and the output, $y(t)$, is taken directly from the third state. Each integrator/summer in Fig. 7 was realized with the three input transconductance-C circuit shown in Fig. 8, where the circuitry for each input transconductance stage is expanded in the dotted box. Here, cross-coupled differen-

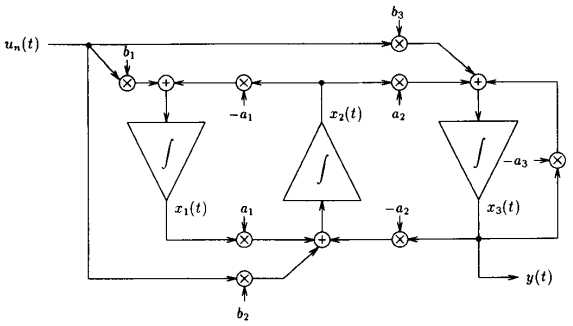


Fig. 7. Block diagram of the adjoint of a third-order orthonormal ladder filter.

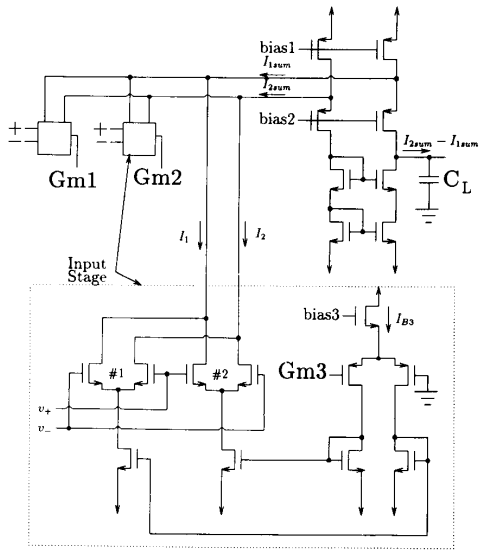


Fig. 8. Circuitry for a 3-input voltage-controlled transconductance-C integrator.

tial pairs enable both positive and negative transconductance, G_m, values. Assuming small-signal operation of the differential pairs #1 and #2, one can show that if the voltage at the node labelled Gm3 is positive, the difference output current, $I_2 - I_1$, will be proportional to the input, $v_+ - v_-$. Conversely, for a negative voltage at Gm3, the output will be proportional to $-(v_+ - v_-)$. However, if the voltage at Gm3 is at ground potential, the output currents will be equal resulting in $I_2 - I_1 = 0$ for arbitrary inputs. The output currents, I_1 and I_2, of each input transconductance stage are summed and reproduced at the output of a high-output-impedance stage realized by stacked current mirrors. Finally, capacitor C_L performs the integration on the output current, $I_{2\text{sum}} - I_{1\text{sum}}$, and also acts as a compensating capacitor to maintain amplifier stability. Note that $I_1 + I_2$ equals the constant current I_{B3}, and therefore, varying individual Gm's does not affect the bias current through the output stage. Also note that all transconductance stages are identical, and thus coefficient values are determined solely by varying control voltages rather than transistor sizing. Although this transconductance approach is beneficial for arbitrary transfer functions, for a known transfer function, it would be advantageous to use transistor sizing to obtain close coefficient values. With the latter approach, the control voltages would only need to fine-tune the transfer function. As a final note, it should be mentioned here that since the integrated circuit we fabricated was mainly intended for the verification of the proposed tuning scheme, no effort was made to linearize the transconductance amplifier.

Making use of these transconductance-C integrators to realize the filter structure of Fig. 7 results in the tunable coefficients, a_i and b_i, being directly proportional to the voltages at the Gm nodes. Voltages at these Gm nodes are then varied in order to tune the filter to the desired transfer function where the set of realizable transfer functions is limited by the range over which the transconductance, G_m, values can be varied. For our prototype, the adaptive algorithm used to adjust these coefficients is the LMS algorithm [14] which attempts to minimize the mean-squared value of the error signal, $e(t)$. To describe the LMS algorithm, consider the case where coefficient b_1 is being adjusted. In this case, the coefficient b_1 is updated according to the following formula:

$$b_1(t) = \int_0^t 2\mu e(\tau)\phi_{b_1}(\tau)\, d\tau \tag{8}$$

where μ is a positive constant that controls the rate of adaptation, and:

$$e(t) = \delta_n(t) - y(t)$$
$$\phi_{b_1}(t) = \frac{\partial y}{\partial b_1}(t). \tag{9}$$

The purpose of using the error, $e(t)$, and gradient, $\phi_{b_1}(t)$, signals may be intuitively interpreted as follows. The error simply determines if y should be increased or decreased in order to match δ_n, while the gradient, ϕ_{b_1}, determines in which direction the coefficient b_1 should be steered. For example, a positive error, $\delta_n > y$, implies that y should be increased. Therefore, to increase y, b_1 should be increased if the gradient is positive but decreased if the gradient is negative.

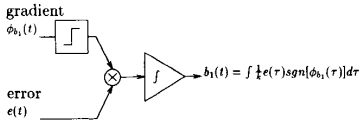


Fig. 9. Block diagram of the LMS coefficient update algorithm.

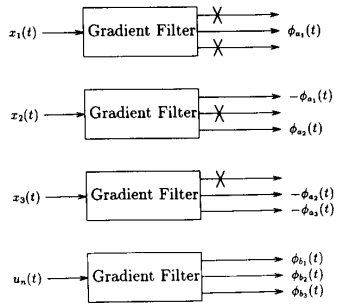


Fig. 10. Block diagram showing the generation of gradients for the third-order filter.

Fig. 9 illustrates a block diagram of the LMS coefficient update procedure. The rate at which the coefficient b_1 is updated is controlled by the step size, μ, which is realized by the integration time constant, k. Note that in this block diagram, only the sign of the gradient signal is used in order to simplify the multiplication circuitry [20].

The error signal is simply obtained by taking the difference between the reference signal and the output of the tunable filter. However, obtaining the gradients is more involved: for the third-order case, the gradients for the coefficients of $\boldsymbol{A}$ and $\boldsymbol{b}$ can be obtained from the output states of four identical state-space filters [15]. We will refer to these extra filters as *gradient filters*. The gradient filter is the transpose or adjoint of the tunable filter and its coefficients are obtained from the coefficients of the tunable filter as follows:

$$\boldsymbol{A}_{\text{grad}} = \boldsymbol{A}^T$$
$$\boldsymbol{b}_{\text{grad}} = \boldsymbol{c}. \tag{10}$$

Note that since the gradients are realized by the states of the gradient filter, only the $\boldsymbol{A}_{\text{grad}}$ and $\boldsymbol{b}_{\text{grad}}$ coefficients are necessary.

The gradient signals for the first column of $\boldsymbol{A}$ are the output state signals of a gradient filter whose input signal is the first state signal of the tunable filter, $x_1(t)$. This is shown in Fig. 10. Note that only the second state signal, $\phi_{a_1}(t)$, is used since the other coefficients in the first column of $\boldsymbol{A}$ are zero. Likewise, the gradients for the

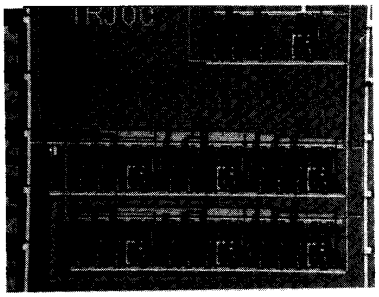


Fig. 11. Integrated chip.

second and third columns of $\boldsymbol{A}$ are the output states of gradient filters with $x_2(t)$ and $x_3(t)$ as the inputs. As illustrated in Fig. 10, only the gradients for the nonzero elements of $\boldsymbol{A}$ are useful. Finally, the gradients $\phi_{b_i}(t)$ are produced by another gradient filter whose input signal is the system input, $u_n(t)$. It should be mentioned here that although the coefficients of the gradient filter are derived from the coefficients of the tunable filter, the matching between these filters is not critical. Since the LMS algorithm follows a path of steepest descent, it is clear that a small change in the path taken will still result in a downward descent to a minimum. Also, an attractive quality of the gradient filter is that it has orthonormal states (since the tunable filter is the adjoint of the orthonormal ladder structure) [18]. Thus the gradients for the coefficients of the $\boldsymbol{b}$ vector will be orthonormal when white noise is applied to the input. Having orthogonal gradients indicates that the coefficients may be updated independently and thus increases the speed of tuning. It has been proven that the adaptation rate of the adaptive linear combiner[2] increases if the gradients are orthonormal [14].

As all four gradient filters are identical, it is possible to build only one gradient filter and multiplex the input and output signals. This multiplexing approach was used in the prototype, and details on the use of multiplexing can be found in [13].

Both the tunable and gradient filters were fabricated using a 3-μm CMOS process, and a photograph of the integrated chip is shown in Fig. 11. A test transconductance cell is shown in the top right corner while the tunable and gradient filters are shown in the middle and bottom of the photograph, respectively. Although simulations indicated the integrated filter could operate up to the megahertz range, the integrating capacitors were kept off-chip in order to experiment with the prototype tuning scheme at lower frequencies. Each filter occupied 0.7 mm^2 and dissipated 3 mW with a $\pm$3-V supply.

For the input PN sequence signal, a repetition length of 15 (i.e., four-bit) was chosen as a trade-off between the complexity of the circuitry for generating the PN sequence and the adaptation performance. The PN sequence was generated by the setup shown in Fig. 12. As discussed previously, the reference signal, $\delta_n(t)$, could be realized by using the system in Fig. 3. However, for more generality, the discrete prototype uses 15 8-bit values stored in an EPROM and an 8-bit DAC as shown in Fig. 13.

[2]The adaptive linear combiner is a common example of an adaptive FIR filter.

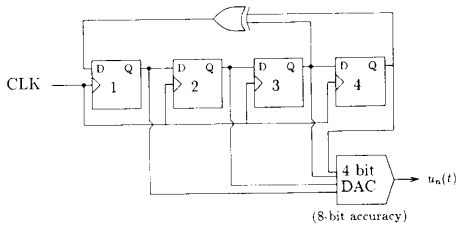


Fig. 12. Four-bit PN sequence circuit.

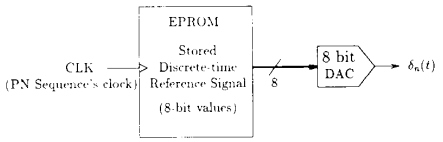


Fig. 13. Block diagram of the reference signal generator.

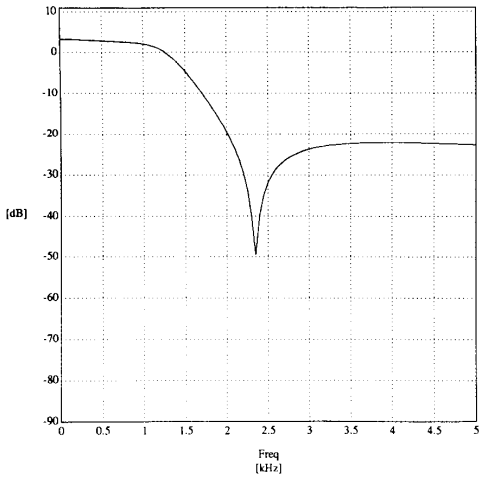
VII. Experimental Results

For testing, it was desired to tune the integrated filter to the poles and zeros given in (1) with the passband edge denormalized to 1 kHz. This desired transfer function is shown in Fig. 14(a). The input and reference sequences were repeated at a rate equivalent to 2/3 kHz. The gradient signals were multiplexed at a rate of 100 Hz and the time constants of the LMS integrators were set to 1 s.

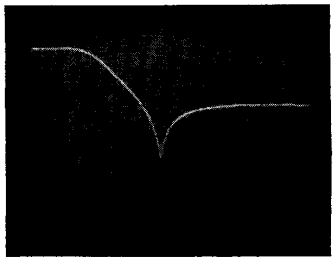
Within 10 to 60 s after power-up, the integrated filter was tuned to the transfer function shown in Fig. 14(b). In order to maintain stability, the minimum time constant of the LMS algorithm integrators is related to the filter's poles. Therefore, it is expected that if the passband edge was at 1 MHz, the adaptation time would be under 60 ms. As well, in this prototype, the initial transfer function of the tunable filter is arbitrary (which accounts for the large variation in tuning speed). By having the initial transfer function closer to the desired one, the tuning speed could be further improved.

VIII. Conclusions

An adaptive technique, based on the traditional model-matching configuration, was presented for tuning integrated continuous-time filters. Whereas present approaches tune only the poles, this adaptive approach tunes all the poles and zeros, thereby allowing a closer match to the desired transfer function. Simulation and experimental results, showing the viability of the proposed approach, were presented. The experimental proto-



(a)



(b)

Fig. 14. Tuned transfer function with a four-bit PN sequence input and 15 reference samples. (a) Simulation results. (b) Experimental results. Vertical scale = 10 dB/div; horizontal scale = 0.5 kHz/div.

type consisted of an integrated third-order filter and off-chip circuitry realizing the adaptive tuning system.

In terms of nonideal effects, dc offsets and parasitic poles appear to be the most critical ones. It has been shown that the effects of dc offsets appearing at the LMS integrators cause a coefficient mismatch and formulas have been derived to determine the effect of these offsets [20]. We are currently developing circuit techniques to reduce the effects of dc offsets. With respect to parasitic poles in the integrators of the filter, this tuning method has the potential for partially accounting for these poles through some innovative circuit and system design.

REFERENCES

[1] 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.
[2] R. Schaumann and M. A. Tan, "The problem of on-chip automatic tuning in continuous-time integrated filters," in *Proc. 1989 IEEE Int. Symp. Circuits and Systems*, pp. 106–109, Feb. 1989.
[3] C. S. Park and R. Schaumann, "Design of a 4-MHz analog integrated CMOS transconductance-C bandpass filter," *IEEE J. Solid-State Circuits*, vol. 23, pp. 987–996, Aug. 1988.
[4] F. Krummenacher and N. Joehl, "A 4-MHz CMOS continuous-time filter with on-chip automatic tuning," *IEEE J. Solid-State Circuits*, vol. 23, pp. 750–758, June 1988.
[5] K. S. Tan and P. R. Gray, "Fully integrated analog filters using bipolar-JFET technology," *IEEE J. Solid-State Circuits*, vol. 17, pp. 814–821, Dec. 1978.
[6] H. Khorramabadi and P. R. Gray, "High-frequency CMOS continuous-time filters," *IEEE J. Solid-State Circuits*, vol. 19, pp. 939–948, Dec. 1984.
[7] K. W. Moulding, J. R. Quartly, P. J. Rankin, R. S. Thompson, and G. A. Wilson, "Gyrator video filter IC with automatic tuning," *IEEE J. Solid-State Circuits*, vol. 15, pp. 963–968, Dec. 1980.
[8] K. Fukahori, "A bipolar voltage-controlled tunable filter," *IEEE J. Solid-State Circuits*, vol. 16, pp. 729–737, Dec. 1981.
[9] V. Gopinathan, Y. P. Tsividis, K. S. Tan, and R. K. Hester, "Design considerations for high-frequency continuous-time filters and implementation of an antialiasing filter for digital video," *IEEE J. Solid-State Circuits*, vol. 25, pp. 1368–1378, Dec. 1990.
[10] Y. P. Tsividis, M. Banu, and J. Khoury, "Continuous-time MOSFET-C filters in VLSI," *IEEE J. Solid-State Circuits*, vol. 21, pp. 15–30, Feb. 1986.
[11] Y. Tsividis, "Self-tuned filters," *Electron. Lett.*, vol. 17, pp. 406–407, June 1981.
[12] Personal communication with W. M. Snelgrove, 1989.
[13] K. A. Kozma, "Tuning integrated continuous-time filters using an adaptive technique," M.A.Sc. Thesis, Univ. Toronto, Aug. 1990.
[14] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[15] 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.
[16] 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.
[17] H. Taub and D. Schilling, *Digital Integrated Electronics*. New York: McGraw-Hill, 1977.
[18] D. A. Johns, W. M. Snelgrove, and A. S. Sedra, "Orthonormal ladder filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 337–343, Mar. 1989.
[19] D. A. Johns, "Analog and digital state-space adaptive IIR filters," Ph.D. dissertation, Univ. Toronto, Mar. 1989.
[20] D. A. Johns *et al.*, "Continuous-time LMS adaptive recursive filters," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 769–778, July 1991.

Karen A. Kozma (S'88) received the B.A.Sc. and the M.A.Sc. degrees in electrical engineering from the University of Toronto in 1988 and 1990, respectively. She is currently working towards the Ph.D. degree at the same university in the area of high-frequency continuous-time filters.

Adel S. Sedra (M'66–SM'82–F'84), for a photograph and biography, please see page 778 of the July issue of this TRANSACTIONS.

David A. Johns (S'81–M'89), for a photograph and biography, please see page 777 of the July issue of this TRANSACTIONS.