Analog and Digital

State-Space Adaptive IIR Filters

by

David A. Johns

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy in the Department of Electrical Engineering.

University of Toronto

March, 1989

Abstract

Adaptive recursive filters are often implemented using direct-form realizations. In this thesis, adaptive algorithms are presented for state-space systems so that the performance of various filter structures may be investigated. Through the use of simulations, it was found that much faster adaptation rates and much improved round-off noise performance may be obtained using structures other than direct-form when final pole locations can be estimated. Since the resulting algorithms are gradient-based, where the gradient signals are obtained as the output of additional filters, both digital and *analog* adaptive recursive filters can be realized. A new *orthonormal ladder filter structure* is presented which has some properties making it attractive for analog adaptive filtering. Specifically, the structure is derived from a singly-terminated LC ladder and has the properties that it is always scaled for optimum dynamic range and its integrator outputs are orthogonal when white noise is applied to the system input. To demonstrate the practicality of analog adaptive recursive filters using the methods in this thesis, experimental results from a discrete prototype are given. As well, the design details and experimental results for a monolithic realization of a continuous-time programmable filter is presented, thus showing the feasibility of practical fully integrated analog adaptive filters. Finally, the effect of DC offsets present in analog implementations is investigated and formulae derived so that the these effects can be estimated and reduced.

Acknowledgments

In writing this thesis, I had the privilege of working and becoming friends with my two supervisors, Adel Sedra and Martin Snelgrove. I wish to thank Adel Sedra for his guidance, support, and wisdom and Martin Snelgrove for his insights, unconventionality, and brilliance. In addition to my two supervisors, many discussions with different people have helped in the development of this thesis. In particular, I wish to thank Gord Roberts, Frank Kschischang, Richard Schreier, and Prof. Bruce Francis. On a broader scope, I appreciate the friendships and interactions made with the electronic and computer group graduate students.

On non-technical issues, two other people in my life have added much to this thesis. My son Christopher created many sleepless nights allowing useful night computer sessions, but more importantly, he has re-introduced the world with a delightful curiosity. Finally, the patience, understanding and love from my wife, Cecilia, has made what seemed an insurmountable task into an enjoyable endeavor.

TABLE OF CONTENTS

Chapter 1

Introduction

1.1. Motivation

Adaptive filters have become an important tool for system designers. Presently, adaptive filters are used as channel equalizers in high speed modems, echo cancellers on telephone lines, and a variety of other applications. In fact, without the use of adaptive filters, the performance of many systems would certainly be degraded. This degradation would mainly be a result of the time varying characteristics found in many engineering problems.

A block diagram of an adaptive filter is shown in figure 1.1. There are two inputs to this

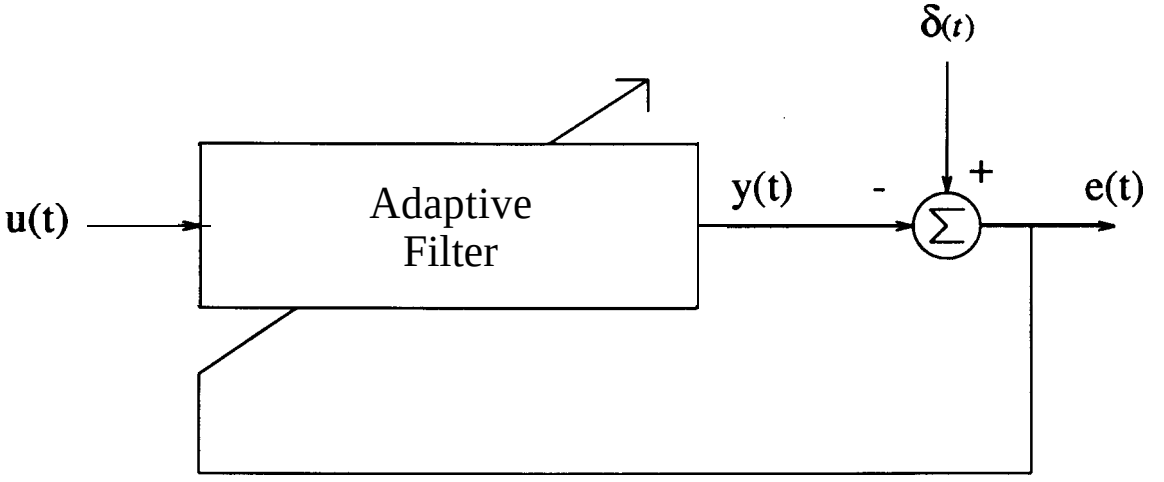


Figure 1.1: An adaptive filter.

system, u(t) and $\delta(t)$, where $\delta(t)$ is often referred to as a ***reference signal***. Qualitatively, the adaptive filter minimizes some measure of the error signal, e(f). In this way, the output of the adaptive filter, $y(t)$, becomes similar to the reference signal, $\delta(t)$, and hence the term ***reference signal***. *Note* that if the adaptive filter could adapt its output instantaneously such that the error signal is always zero, then the filter output $y(t)$ would be equal to $\delta(t)$. However, this is not effect that one wants to obtain with an adaptive filter. In fact, although the adaptive filter minimizes the error signal, in many applications the error signal will not and should not go to zero. The goal of an ideal adaptive filter is to force the error to be at a minimum only to the extent that a fixed linear filter could also achieve assuming the charateristics of the input signals were time-invariant. To achieve this goal, the adaptive filter's transfer function varies slowly in comparison to the signals $u(t)$ and $\delta(t)$. In this way, once the adaptive filter converges, only the part of the signals in u(f) and $\delta(t)$ which are related by a linear transfer function are subtracted to reduce the error signal. To illustrate this point, an example is given below of an echo canceling application where the error signal should not go to zero. Finally, it should be noted that in an adaptive filter system, the output is usually taken as either the error signal, $e(t)$, or the filter output, y (t). Alternatively, in some applications, coefficient values describing the transfer function of the adaptive filter are the desired output.

Let us now consider one application of an adaptive filter to understand how one applies this technology. A block diagram showing the application of an adaptive filter to an echo cancellation problem is shown in figure 1.2. In this example, a 2-4 wire hybrid is used to convert two pairs of wires carrying the receive and talk signals (referenced to telephone set B) to a single pair of wires having both talk and receive signals on it. With an ideal 2-4 wire hybrid, there would be no need for an adaptive filter. However, in actual implementations, there is always some amount of receive signal which leaks through the hybrid on to the talk signal. This leaked receive signal

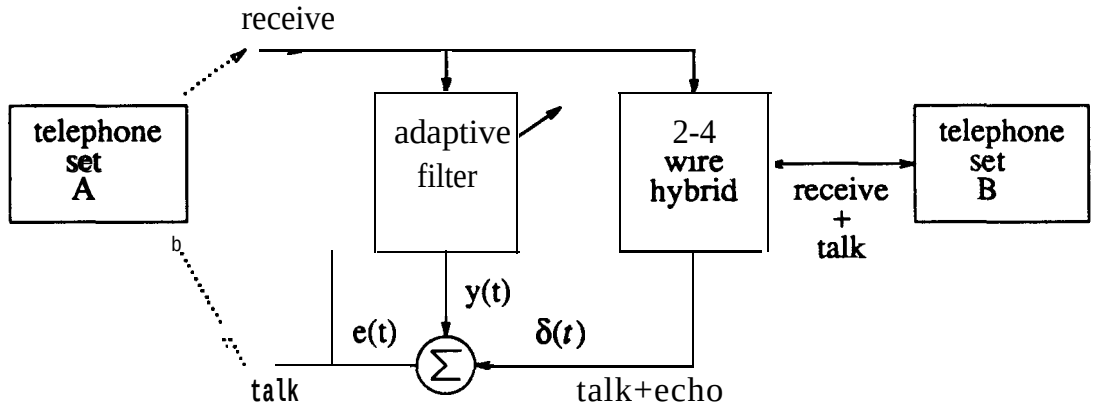


Figure 1.2: An adaptive filter used in an echo cancellation application.

is referred to as an echo signal since the speaker on telephone set A will hear their own voice after some delay through the system. Using an adaptive filter, as shown in figure 1.2, one can reduce the amount of echo. The echo is reduced by the adaptive filter attempting to match the filter output, $y(t)$, with the "talk + echo" signal. Since the signal $y(t)$ is a filtered version of the receive signal (hopefully uncorrelated with the talk signal), the only way to minimize the error signal, e(t), is to match y(t) to the echo signal. In this way, the talk signal is sent on with a reduced amount of echo signal. Note that for this example, one could actually replace the adaptive filter with a linear filter if one knew the characteristics of the 2-4 wire hybrid. However, this approach is not practical as the characteristics of the hybrid are not trivial to determine and change with different telephone connections. For further applications of adaptive filters, the reader is referred to [Widrow and Stearns, 1985].

From the above example, we see that an adaptive filter can be thought of as a linear filter which changes its transfer function over time in order to minimize some error criterion. Using

the fact that a transfer function can be described in the frequency domain by poles and zeros, one can classify adaptive filters into two types; Infinite-Impulse-Response (IIR) and Finite-Impulse-Response (FIR) adaptive filters. Adaptive FIR filters change only the zeros of the transfer function while adaptive IIR filters change both the poles and zeros of the transfer function'. Presently, most adaptive filter implementations consist of the FIR type due to the speed of convergence in finding a minimum and a guaranteed convergence to the minimum error. However, to achieve a satisfactory performance with an adaptive FIR filter, a high order filter is often required. In many applications, this order can often be significantly reduced by using an adaptive IIR filter where both the poles and zeros of the filter's transfer function are adjusted. However, there are problems associated with adaptive IIR filters such as converging to a local minimum and ensuring that the filter remains stable. Nevertheless, because of the reduction in filter order, there is considerable interest in understanding and developing practical adaptive IIR filters. In fact, although the theory behind adaptive IIR filters is not yet well established, there are some applications where adaptive IIR filters are now being applied [Eriksson and Allie, 1988]

As well as classifying adaptive filters into IIR or FIR types, one can also classify adaptive filters into two main implementation technologies; analog and digital. Digital implementations are the most common method of adaptive filter realization where digital signal processing blocks are used to realize the necessary programmable filters. This technology is especially suited for programmable filters since filter coefficients realized with random-access-memory (RAM) are easily changed. However, the use of digital signal processing blocks limits the types of applications to those that can be efficiently realized with digital technology. Specifically, it is well known that analog filters can process much higher frequencies than digital filters. As well, in

[1] Note that the transfer function of the filter in an adaptive FIR filter may be of the IIR type but only the zeros

applications where no digitization is necessary except for filtering, analog implementations require much less silicon area than the equivalent digital systems. Thus, there are applications where analog adaptive filters are used to meet system specifications.

In table 1.1, a summary of the present theoretical base and implementation usage for the different adaptive filter types is presented. This table indicates that the digital adaptive IIR filter techniques are just starting to mature and that analog adaptive IIR filters are only very recently being investigated. In fact, the analog adaptive IIR filtering results so far arc given only for reasearch implementations IMikhael and Yassa, 1982]. Thus, the main motivation of this thesis is to find a practical implementation technique for creatinganalog adaptive IIR filters.

1.2. State-of-the-art review

Historically, one of the first digital adaptive IIR filter algorithms in the signal processing literature was presented in a 1975 publication mite, 1975]. This algorithm used a gradient

		Adaptive filter type	
		FIR	IIR
Technology	Digital	Well established theory and most common implementation method	Growing theory base and some implementations
	Analog	Well established theory and mostly high speed implementations	Very little theory and only few research implementations

Table 1.1: The theory base and usage of different adaptive filter types.

based approach applied to a direct-form filter in order to update the filter coefficients so that an error minimum could be located. However, one of the main disadvantages of this algorithm is that many computations are required to compute the necessary gradients. In 1976, a new algorithm was presented [Feintuch, 1976] that significantly reduced the computations required to adapt an IIR direct-form filter. Although this algorithm has been criticized [Johnson and Larimore, 1977] [Widrow and McCool, 1977] and examples found where it does not converge [Larimore et al., 1980], it has recently been used in an industrial application because of its simplicity [Eriksson and Allie, 1988]

A different approach to adapting IIR direct-form filters was presented in [Larimore et al., 1980] where hyperstability theory is applied. This hyperstability approach can guarantee convergence to a global minimum if a certain strictly positive real condition is met. Unfortunately, ensuring that this condition is met is not a trivial matter.

The one common point between all the algorithms discussed so far is that they have been derived assuming a direct-form digital filter. Since this type of filter is known to be very poor in analog implementations, it would be desirable to find algorithms which do not rely on this structure. In the digital literature, algorithms exist for adapting the lattice IIR structure [Parikh et al., 1980] [Ayala, 1982], however, these algorithms require significant calculations to compute gradients and it is not clear how one would convert these algorithms to an analog equivalent. As well as the lattice structure, an algorithm was presented for adapting digital biquad IIR filters [Martin and Sun, 1987]. While the work was performed independently, the algorithm in this publication is quite similar to this author's approach in that sensitivity filters are used to obtain the necessary gradients to adapt structures other than direct-form. However, whereas this author's approach applies to state-space structures, the algorithm in [Martin and Sun, 1987] is intended

for biquad structures.

With respect to analog adaptive filtering techniques, implementations presently exist for high frequency applications [Qureshi, 1985] [Treichler et al., 1987]. As well, papers have recently been published concerning new analog realizations of adaptive filters [Lev-Ari et al., 1987]. However, the usual approach has been to use analog delay lines and adapt only the zeros of the transfer function. With the use of analog delay lines, much of the work in the digital FIR literature can be readily applied; however, integration of a complete adaptive system becomes quite difficult. As well, adapting only the zeros in an analog system can lead to high order systems, as discussed above. While in digital systems, orders as high as 200 can be realized, these high orders are often too large for an analog implementation.

With respect to analog IIR filters, this author has seen only one publication which presents a possible technique for implementation [Mikhael and Yassa, 1982]. In this publication, both the poles and zeros are adapted using a sequential-linear-search algorithm that can be described as simply changing a filter coefficient and then measuring the change in the output error RMS voltage. If the error voltage decreases, the coefficient is left at that changed position while if the error increases then the coefficient is changed in the opposite direction. Although this is a simple technique, there is a serious drawback in that small changes in error RMS voltages must be observed while the absolute value of the error voltages might be large. Thus, the algorithm requires an extremely accurate RMS measurement of the error signal which is difficult to obtain.

1.3. Outline of thesis

In chapter 2, necessary background material is presented. Notation and terms are defined as well as presenting some adaptive filter theory. Also in this chapter, a brief introduction to state-space theory is given.

Chapter 3 describes a filter structure having some useful properties for fixed or adaptive filtering. This new structure was first obtained in ISnelgrove, 1982] by performing a Gram-Schmidt orthogonalization procedure on companion-form filters. Although no proof was given, it was conjectured that the structure would always have orthonormal states. In chapter 3 of this thesis, it is shown that this conjecture is true and design equations and sensitivity results are given showing the ease of design and the usefulness of this new structure.

State-space adaptive IIR filter algorithms are presented in chapter 4. It is shown in this chapter that general state-space structures can be adapted using extra filters to obtain gradient signals. It is also shown that one can adapt a single column or row of a state-space filter and therefore reduce the number of extra gradient filters to one. These single row or column adaptive filters are shown to have superior convergence properties as compared to direct-form filters in oversampled applications where one can estimate the final pole locations. At the end of chapter 4 is an appendix giving a design procedure to obtain a "quasi-orthonormal" digital filter for oversampled applications.

Chapter 5 presents experimental results for a discrete prototype which demonstrates that the algorithms in chapter 4 can successfully be converted to the analog domain. As well, the design details for a monolithic programmable continuous-time filter are given along with experimental results for a fabricated device.

It was found during experimentation with the discrete prototype that DC offsets are a serious concern with analog adaptive filters. In chapter 6, formulae are developed giving the coefficient error and excess mean squared error due to DC offsets. Also in this chapter are some experimental results, obtained on the discrete prototype, verifying the usefulness of these formulae.

Finally, conclusions and future work are presented in chapter 7.

Chapter 2

Background Theory

This chapter will present the necessary background material for a proper understanding of the remaining chapters. The notation used throughout this thesis will be described in the first section with the following section presenting some signal processing definitions concerning the terms expectation, correlation and norm. The important aspects of adaptive filter theory relating closely to this thesis work are presented in section 2.3. Finally, section 2.4 will present a review of state-space theory where, again, only the material closely related to this thesis work will be described.

2.1. Notation usage

In order that the reader can more readily follow the material presented in this thesis, understanding the notation usage is important. This notation has representations for continuous and discrete time functions and their transforms as well as vectors and matrices.

The notation usage is the following: Continuous-time functions are represented by lowercase letters and are functions of a variable (eg. x(r)). The Laplace transform of a function is written using uppercase letters and is also a function of a variable (eg. X(s)). With a similar lowercase and uppercase convention, discrete-time functions and their Z-transform are represented (eg. $x(n)$ and X(z)). Although continuous and discrete time functions and their transforms use the same notation, no confusion should arise since only one domain is assumed at any time throughout this thesis. A set of functions is often written as a vector which is represented using a bold typeface (eg. $\mathbf{x}(t)$). Vectors and matrices are also represented using a

bold typeface. As well, vectors are represented using lowercase letters (eg. **c**) while uppercase letters are used for matrices (eg. **A**). Table 2.1 below summarizes the above rules using examples.

$x(t)$	---	continuous-time function
$x(n)$	---	discrete-time function
$X(s)$	---	Laplace transform of x (t)
X(z)	---	Z-transform of $x(n)$
$\mathbf{x}(t)$	---	vector of continuous-time functions
$\mathbf{X}(s)$	---	vector of frequency functions
c	---	vector
A	---	matrix
$x_i(t)$	---	the i'th element of vector $\mathbf{x}(t)$
c_i	---	the i'th element of vector c
A_{ij}	---	the element in the i'th row and j'th column of **A**

Table 2.1: Examples of notation usage.

2.2. Expectations, correlations and norms

Throughout this thesis, concepts such as expectation, correlation and norms of signals are important. As well, signals may be of one of two types: finite energy or finite power. In this thesis, finite energy signals result from system impulse responses and are therefore deterministic signals. However, power signals, in this thesis, result from a noise source being applied to a system and are therefore non-deterministic signals. Specifically, power signals are node signals of a system when a noise source is applied. We now proceed to give some meaning to the concepts of expectation, correlation and norm. Note that all time signals are assumed to be real valued.

2.2.1. Finite power signals

In this section, all signals are assumed to be finite power signals. The aforementioned concepts will be defined for the ergodic case where time averages may be substituted for ensemble averages. It should be pointed out that many signals involved in an adaptive filter will not be ergodic even with ergodic inputs since we are dealing with a time-varying system. However, in practice, if the system is varying slowly then the short term average of a signal in a time-varying system at time t_1 will approximate the long term average of the equivalent signal in a time-invariant system whose system coefficients match the time-varying system coefficients at time t_1. So, although the definitions to be presented are not strictly correct for the time-varying case, they still give some physical meaning to the concepts of expectation, correlation and norms.

We write the expectation of the signal x as $E[x]$. ***In the*** discrete-time case, the expectation is defined as

$$E[x] \equiv \lim_{N\to\infty} \frac{1}{2N+1} \sum_{-N}^{N} x(n) \tag{2.1}$$

whereas for the continuous-time case

$$E[x] \equiv \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)dt \tag{2.2}$$

The inner product or correlation between two signals x and y is written as $E[xy]$. For the discrete-time case, this correlation is defined as

$$E[xy] = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{-N}^{N} x(n)y(n) \tag{2.3}$$

and in the continuous-time domain

$$E[xy] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)y(t)dt \tag{2.4}$$

When $E[xy] = 0$, the two signals are said to be orthogonal or uncorrelated.

Another useful definition is the norm of a signal. The norm of the signal x is written as $\|x\|_P$ where the subscript P denotes that the norm is defined for a power signal. This definition should not be confused with the general p'th norm of a signal. This power norm is defined as

$$\|x\|_P \equiv \left[E[x^2]\right]^{\frac{1}{2}} \tag{2.5}$$

and is simply the root mean squared (RMS) value of the signal. The norm squared is also referred to as the mean squared value. Note that this norm assumes the power signal is not a finite-energy signal.

2.2.2. Finite energy signals

In this thesis, we also require the concept of correlation between transfer functions or, equivalently, finite energy signals. These definitions are only required in the continuous-time domain and therefore will only be defined in that domain.

First, we shall define the norm used for finite energy signals and from that norm, define the concept of correlation between two transfer functions. We define these energy norms in such a way as to be useful in predicting power norms.

Consider the system shown in figure 2.1 where the input is a white noise signal with a spectral density of $1\frac{V^2}{rad/s}$. Choosing to define the norm, $\|F(s)\|_E$, of the transfer function $F(s)$ as the output RMS voltage resulting from this white noise input signal, one can use the Wiener-Khinchine theorem [Ziemer and Tranter, 1976] to show that[1]

$$\|F(s)\|_E^2 = \int_{-\infty}^{\infty} |F(j\omega)|^2 d\omega \tag{2.6}$$

This norm definition leads naturally to the following correlation definition. The correlation

[1] This integral is finite for the practical case of stable, strictly proper rational $F(s)$.

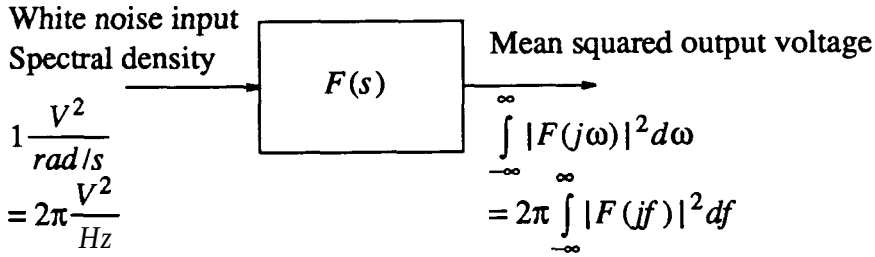


Figure 2.1: System used to define energy signal norm.

between two transfer functions, $F_1(s)$ and $F_2(s)$, written as $<F_1(s),F_2(s)>$ is defined as

$$<F_1(s),F_2(s)> = \int_{-\infty}^{\infty} F_1(j\omega)\bar{F}_2(j\omega)d\omega \tag{2.7}$$

which by Parseval's relation, equals the inner product in the time domain given by

$$<f_1(t),f_2(t)> = 2\pi\int_0^{\infty} f_1(t)f_2(t)dt \tag{2.8}$$

where $f_1(t)$ and $f_2(t)$ ***are the*** impulse responses of the two transfer functions. As before, if $<f_1(t),f_2(t)>$ equals zero, we say the two transfer functions are orthogonal.

Finally, we would like to determine what the correlation between two transfer functions implies. Towards this goal, consider the system shown in figure 2.2 where a noise signal, u(f), is used as the input to two systems with impulse responses $f_1(t)$ and $f_2(t)$. ***The*** correlation between the two output power signals, $x_1(t)$ and $x_2(t)$, is found from

$$E[x_1x_2] \equiv \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x_1(t)x_2(t)dt \tag{2.9}$$

$$= \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} [f_1(t)\otimes u(t)][f_2(t)\otimes u(t)]dt \tag{2.10}$$

where $\otimes$ denotes convolution.

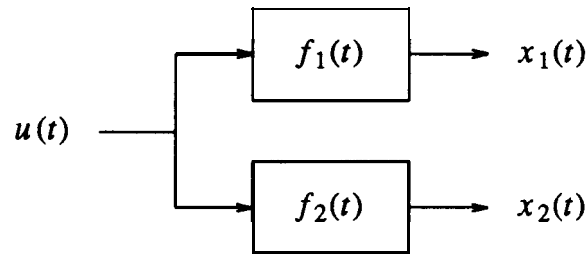


Figure 2.2: System used to show that white noise into orthogonal transfer functions creates orthogonal output signals.

Using some manipulations between expectation and correlations [Ziemer and Tranter, 1976], the following equation can be derived.

$$E[x_1 x_2] = \int_0^\infty f_2(\tau)[f_1(\tau) \otimes R_u(\tau)]d\tau \tag{2.11}$$

where $R_u(\tau)$ is the autocorrelation of the input signal, u(t). In the case where the input signal has an autocorrelation function equal to an impulse (white noise), the above correlation reduces to

$$E[x_1 x_2] = \int_0^\infty f_2(\tau) f_1(\tau) d\tau = \frac{1}{2\pi} < f_1(t), f_2(t) > \tag{2.12}$$

Thus, orthogonal finite power signals can be obtained from a white noise input signal and orthogonal transfer functions.

2.3. Some adaptive filter theory

This section will present some background theory concerning adaptive filters. Only the basic theory that pertains to the material in this thesis will be presented. For a more thorough description on adaptive filter theory, the reader is referred to IWidrow and Stearns, 1985] and

[Treichler et al, 1987].

A block diagram of a discrete-time adaptive filter is shown in figure 2.3. The filter is programmed by adjusting its filter coefficients, $\{p_i\}$. The output of the programmable filter is subtracted from a reference signal, $\delta(n)$, to create an error signal, e (n). The adaptive algorithm uses the error signal and filter states to adjust the filter coefficients in such a way as to minimize the norm of the error signal. This error norm can be thought of as an error performance surface mapped out by varying all the filter coefficients. Thus, the adaptive algorithm attempts to find a minimum in the error performance surface by adjusting the filter coefficients.

An approach to finding a minimum in a performance surface is to use the method of steepest descent. Applying this method, each filter coefficient is updated independently and, as

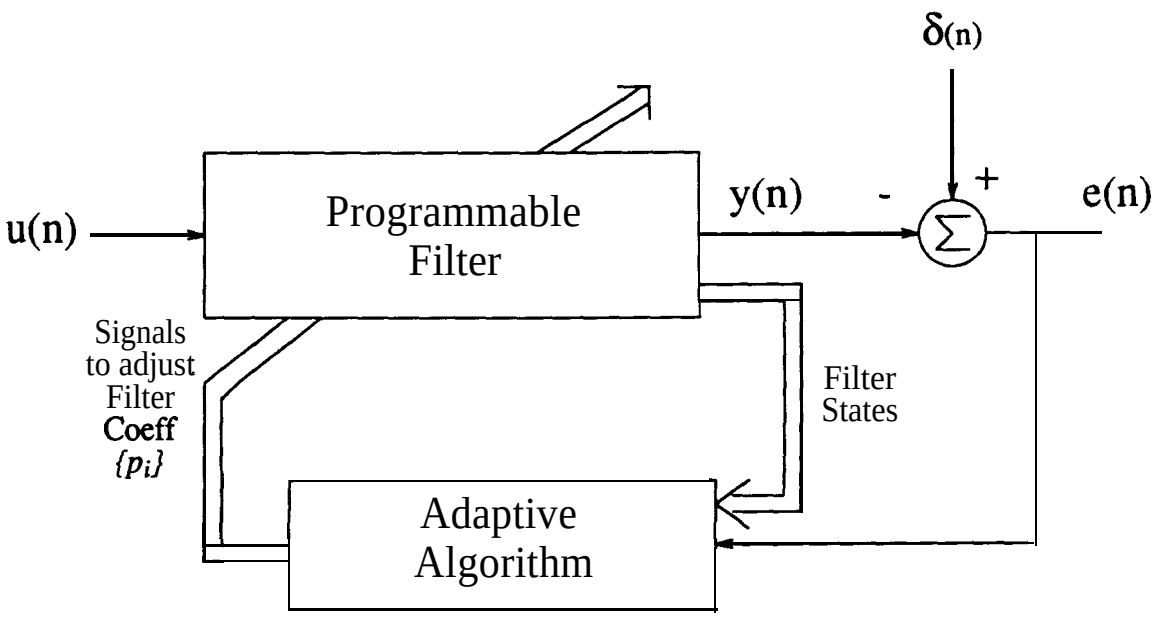


Figure 2.3: Adaptive filter block diagram.

the name suggests, the surface is traversed in the direction of steepest descent. To find a minimum in the error performance surface, the steepest descent update equation for the i'th coefficient is written as:

$$p_i(n+1) = p_i(n) - \mu \frac{\partial \left[E(e^2) \right]}{\partial p_i} \tag{2.13}$$

where μ is a small positive step size parameter which controls the rate of convergence. Unfortunately, it is usually difficult to obtain the partial derivative term involving the mean squared error. To circumvent this problem, the least-mean-squared (LMS) algorithm was developed [Widrow and Hoff, 1960]. With this approach, the instantaneous error squared signal is used to approximate the mean squared error. Substituting in the formulae for the error signal, $e(n) = \delta(n) - y(n)$, and using the fact that the reference signal is not a function of the parameter, p_i, the following LMS update equation is obtained.

$$p_i(n+1) = p_i(n) + 2\mu e(n) \frac{\partial y(n)}{\partial p_i} \tag{2.14}$$

Although in this equation there are no explicit expectation operators, the expectation operation is performed over time during adaptation of the parameter p_i assuming a small step size, μ. Thus, although the instantaneous gradient may often point in the wrong direction, on average it will point in the correct direction and the adaptation path of the coefficients will follow the line of steepest descent.

Although this thesis will present methods to adapt both the poles and zeros of an adaptive filter, let us look at the well-known special case where only the zeros of a transfer function are adapted using a linear combiner as shown in figure 2.4. By looking at the reduced problem of adapting only zeros, some insight can be gained regarding the adaptive LMS algorithm.

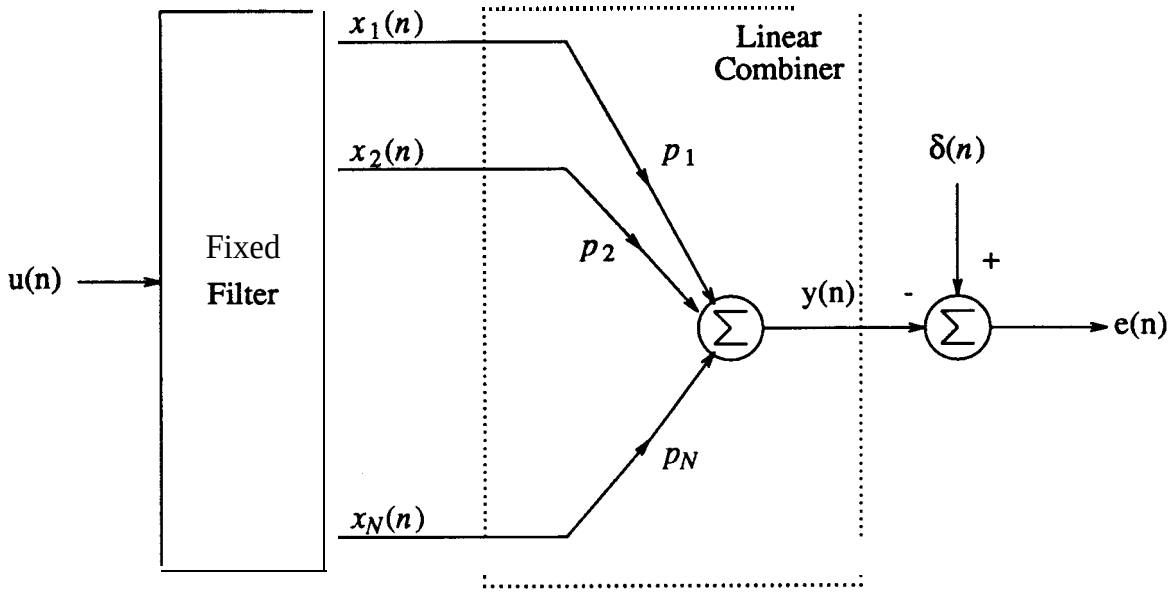


Figure 2.4: A linear combiner adaptive filter

The programmable filter in figure 2.4 consists of both the fixed filter and linear combiner where the fixed filter has one input and N independent outputs. In many implementations, the fixed filter is simply a tapped delay line resulting in a programmable filter which is simply an FIR transversal filter. One reason for the great interest in adaptive linear combiners is that the error performance surface is quadratic and therefore has only one minimum which is easily found using the LMS algorithm The output of the programmable filter, y (n), can be written as

$$y(n) = \sum_{i=1}^{N} p_i x_i(n) \tag{2.15}$$

Therefore, the partial derivative of y (n) with respect to p_i is simply $x_i(n)$. This fact leads to the following update equation for the coefficient p_i.

$$p_i(n+1) = p_i(n) + 2\mu e(n)x_i(n) \tag{2.16}$$

Note that all the signals in this equation are readily available as system signals in the adaptive linear combiner and that there are only multiplications and additions required. This fact is the reason that the LMS algorithm is the most popular adaptive filter algorithm.

The LMS algorithm can be extended into the analog domain quite naturally [Widrow et al, 1967]. Assuming an analog adaptive linear combiner, the update equation for the multiplying coefficient p_i is simply

$$p_i(t) = 2\mu \int_0^t e(\tau)x_i(\tau)d\tau \tag{2.17}$$

A tapped analog delay line is often used to obtain an independent set of states $x_i(t)$ from a single input.

2.3.1. Why orthonormal signals are good

An orthonormal set of signals is a set where the norms of all the signals in the set are the same and the correlation between any two different signals is zero. This section will investigate the effect of different sets of input signals on the linear combiner. What will become apparent is that the cross correlations of the input signals, $x_i(n)$, affect the convergence performance of the adaptive filter. To quantify this effect, an input correlation matrix R is defined as

$$\mathbf{R} \equiv E[\mathbf{x}\mathbf{x}^T] = \begin{vmatrix} E[x_1x_1] & E[x_1x_2] & \cdots & E[x_1x_N] \\ E[x_2x_1] & E[x_2x_2] & \cdots & E[x_2x_N] \\ \vdots & \vdots & \ddots & \vdots \\ E[x_Nx_1] & E[x_Nx_2] & \cdots & E[x_Nx_N] \end{vmatrix} \tag{2.18}$$

With this definition, the element R_{ij} equals the correlation between the states $x_i(n)$ and $x_j(n)$. The diagonal elements of R are simply the mean squared values of the states. It is not difficult to show that R is positive semidefinite and therefore all the eigenvalues are greater than or equal to zero. In fact, for most well behaved problems all the eigenvalues are greater than zero. Defining

λ_{min} and λ_{max} to be the minimum and maximum eigenvalues of R, respectively, one can show that these two numbers help in determining the convergence properties of the adaptive filter. Specifically, if the LMS algorithm is modeled as though the true gradient is followed rather than the noisy instantaneous one, it can be shown that the step size parameter, μ, must be bounded to guarantee convergence. This bound is determined by λ_{max} as

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{2.19}$$

The step size μ must be larger than the lower bound in order to adapt while μ must be smaller than the upper bound for the algorithm to remain stable. As well, it can be shown that the minimum eigenvalue determines the time constant, τ, for overall convergence of the coefficients $\{p_i\}$.

$$\tau = \frac{1}{\mu\lambda_{min}} \tag{2.20}$$

Therefore, if one chooses μ to be a fraction, a, of the maximum value which guarantees convergence,

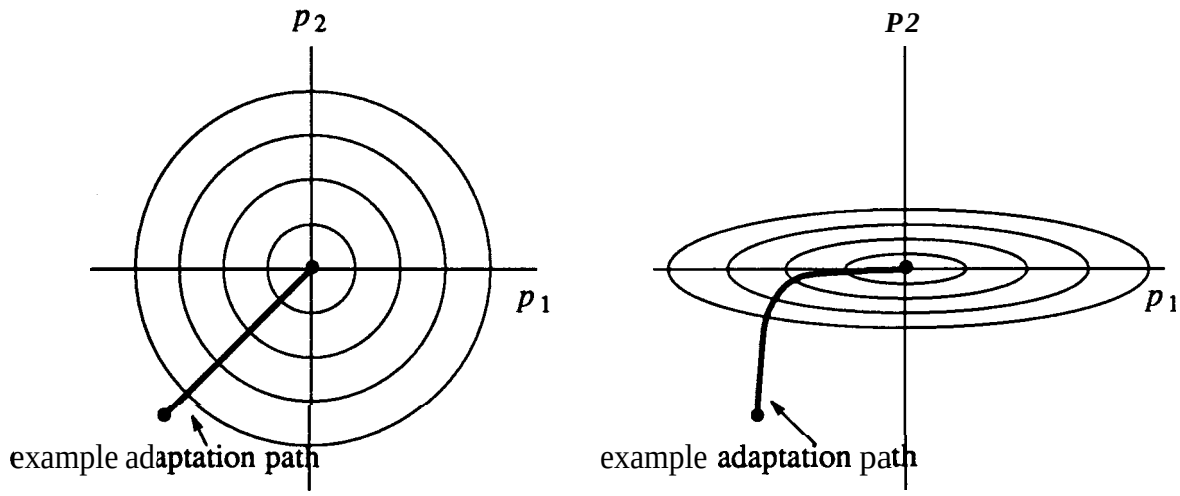
$$\mu = \frac{\alpha 2}{\lambda_{max}} \tag{2.21}$$

then

$$\tau = \frac{1}{2\alpha}\frac{\lambda_{max}}{\lambda_{min}} \tag{2.22}$$

This equation indicates that for a constant proportion α, all the eigenvalues should be equal for the shortest time constant. The identity matrix has such a property. Therefore, if R equals the identity matrix, good adaptation properties are obtained. Note that an input correlation matrix corresponding to the identity matrix implies that the set of input signals constitute an orthonormal set.

For a more physical interpretation of the above results, consider the contour plots of the error performance surfaces shown in figure 2.5. For those readers who are unfamiliar with contour plots, a contour plot is a method of showing a 3-dimensional surface. Two of the axes are in the surface of the page with the third axis coming straight out of the page. The contour lines are simply lines of equal height in the third axis. Therefore the steepest descent path at a point on a contour line is perpendicular to the tangent at that point. As well, a measure of the steepness of the steepest descent path can be determined by the distance that the contour. lines are apart along that path. (This assumes that the contour lines are at regular intervals along the third axis.) In figure 2.5, linear combiners consisting of two inputs are assumed with variable multiplying coefficients p_1 and p_2. The minimum error is obtained when $p_1 = p_2 = 0$ or, equivalently, at the



Orthonormal states Non-orthonormal states

Figure 2.5: Contour plots of two performance surfaces (minimum error is at origins).

origin[2]. For the orthonormal example, the contours are circles while the non-orthonormal case has elliptical contours. In fact, the non-orthonormal example shown has orthogonal states but the norms of the states are not equal. A non-orthogonal example will simply have the ellipses tilted so that the principal axes of the ellipses do not line up with the coefficient axis. What is seen from figure 2.5 is that in the orthonormal example, the gradient always points towards the minimum. Therefore, if a steepest descent algorithm is applied to some starting point where $|p_1|=|p_2|$, both coefficients will approach 0 at the same rate. In contrast, if the same conditions are applied to the non-orthonormal example, the coefficient p_2 will converge must faster than the coefficient p_1. Thus, for this case, the steepness along the p_1 axis dominates the convergence time while the stability of the algorithm is determined by the steepness along the p_2 axis.

Concluding this section, we have shown that an orthonormal set of states is desirable for a good adaptation convergence rate.

2.3.2. Adaptive IIR filters

Adapting the poles of an adaptive filter as well as the zeros adds several complexities to the system design. Some of these complexities will be discussed in this section.

First, in contrast to the adaptive linear combiner, the error performance surface can have many minima. Therefore, a steepest descent algorithm may find a local minimum rather than the global minimum. Fortunately, there appears to be a large class of applications where the performance surfaces of adaptive IIR filters have only one global minimum [Stearns, 1981]. In the above reference, although no proof was found, it is conjectured that if the order of the adaptive

[2] Note that this is a degenerate case where the output signal is 0 when the the coefficients are at the minimum point. However, this case still indicates the benefits of an orthonormal set.

filter is greater than or equal to the system being modeled, then the error performance surface will have one global minimum.

Another concern with adaptive IIR filters is that the algorithm may try to place poles of the programmable filter in the instability region. For this reason, some method is usually required to ensure that the coefficients of the programmable filter are always chosen such that the filter is stable. This can be accomplished by choosing filter structures having simple stability checks.

Finally, it should be pointed out that obtaining the gradients for adaptive IIR filters is not as simple as in the FIR case. In fact, one of the first approaches to adaptive IIR filters white, 1975][Stearns et al, 1976] required approximately N times the computations of the programmable filter (where N is the programmable filter's order) to obtain the gradients of the coefficients. This complexity was significantly reduced in an algorithm where the gradients were approximated [Feintuch, 1976]. Unfortunately, with this gradient approximation there is no guarantee of converging to any minimum. Finally, it should be mentioned that there are other approaches to adaptive IIR filtering than gradient-based approaches. One example is the SHARF algorithm [Larimore et al, 1980] where hyperstability theory is utilized but has limited applications because it relies on a strictly positive real condition which is difficult to guarantee without knowing a great deal about the application.

2.4. State-space theory

If one wishes to build a system with a given linear time-invariant transfer function using only ideal integrators and summing blocks, there are an infinite number of realizations which would result in the given transfer function. However, not all realizations would have the same performance when constructed with actual components. Therefore, we require simple formulae allowing us to investigate the filter's performance when realized with different structures.

State-space system theory has such formulae and is well established in the control and signal processing literature. Thus, using implementations having a close relationship to state-space systems, we can benefit from the wealth of analysis tools available for state-space systems.

2.4.1. State-space system description

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

$$s\mathbf{X}(s) = \mathbf{A}\mathbf{X}(s) + \mathbf{b}U(s) \tag{2.23}$$

$$\mathbf{Y(s)} = \mathbf{c}^T\mathbf{X}(s) + dU(s)$$

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

$$T(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d \tag{2.24}$$

From equation (2.24) above, we can see that the poles of the system are included in the eigenvalues of **A** and therefore are determined by only one system coefficient. However, the zeros of the system are related to all four of the system coefficients.

To obtain more insight into state-space systems, two sets of intermediate-functions need to be defined ISnelgrove and Sedra, 1986]. The first set of functions, **F(s),** is defined as the transfer-functions from the system input to the output of each of the integrators,

$$F_i(s) \equiv \frac{X_i(s)}{U(s)} \tag{2.25}$$

This definition leads to the formula

$$\mathbf{F(s)} = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} \tag{2.26}$$

The second vector of functions, G(s), is defined as the set of transfer functions from the input of

each of the integrators to the output. Thus if we inject a signal $\varepsilon_i(s)$ at the input of the i'th integrator, then

$$G_i(s) \equiv \frac{Y(s)}{\varepsilon_i(s)} \tag{2.27}$$

Using this definition we have

$$\mathbf{G}^T(s) = \mathbf{c}^T(s\mathbf{I}-\mathbf{A})^{-1} \tag{2.28}$$

A transformation that will prove useful throughout the remainder of this thesis creates a new system which exchanges the two sets of intermediate-transfer functions [Jackson, 1970]. Specifically, given a system, $[\mathbf{A},\mathbf{b},\mathbf{c},d]$ with intermediate-transfer functions, $\mathbf{F}(s)$ and $\mathbf{G}(s)$, we can create a new system such that

$$\mathbf{F}_{new}(s) = \mathbf{G}(s) \quad \text{and} \quad \mathbf{G}_{new}(s) = \mathbf{F}(s) \tag{2.29}$$

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

$$\mathbf{A}_{new} = \mathbf{A}^T \quad \mathbf{b}_{new} = \mathbf{c} \quad \mathbf{c}_{new} = \mathbf{b} \quad d_{new} = d \tag{2.30}$$

This result can be easily verified using the formulae in equations (2.26) and (2.28) above. We shall refer to this new system as the transposed system of the original system[3].

2.4.2. Sensitivity equations

Sensitivity formulae relating the change in the transfer-function to changes in the system coefficients will be presented in this section. These formulae can be derived using the formula

$$\frac{dT_{ab}}{dT_{mn}} = T_{am}T_{nb} \tag{2.3 1}$$

where T_{ij} is a transfer-function from point i to point j in a system. An elegant derivation of this formula is presented in ISnelgrove and Sedra, 1986].

[3] The tranposed system is often referred to as the adjoint system in the circuit literature.

Using equation (2.31), it is not difficult to find the following sensitivity formulae relating the change in the transfer-function to changes in the system coefficients.

$$S_{A_{ij}}^{T(s)} = g_i(s) f_j(s) \frac{A_{ij}}{T(s)} \tag{2.32}$$

$$S_{b_i}^{T(s)} = g_i(s) \frac{b_i}{T(s)} \tag{2.33}$$

$$S_{c_i}^{T(s)} = f_i(s) \frac{c_i}{T(s)} \tag{2.34}$$

$$S_d^{T(s)} = \frac{d}{T(s)} \tag{2.35}$$

where $S_p^{T(s)}$ is the classical sensitivity measure defined as

$$S_p^{T(s)} = \frac{\partial T(s)}{\partial p} \frac{p}{T(s)} = \frac{\partial \ln T(s)}{\partial \ln p} \tag{2.36}$$

These simple formulae will prove invaluable to us when an adaptation algorithm for IIR filters is proposed.

2.4.3. Correlation matrices

It is often useful **to know** the correlation between intermediate-functions. Note that these functions are of finite energy if the input is of finite energy and **A** is stable, therefore we use the above definitions of correlation and norm for finite energy signals. We now define two correlation matrices. The first matrix, **K,** is defined as

$$K_{ij} = \langle F_i(s), F_j(s) \rangle \tag{2.37}$$

whereas the second matrix, **W,** is defined as

$$W_{ij} = \langle G_i(s), G_j(s) \rangle \tag{2.38}$$

Note that the diagonal elements of **K are** the squared norms of the intermediate $\mathbf{F}(s)$ functions. Therefore, if **K** equals the identity matrix, I, the intermediate **F** functions constitute an orthonormal set of functions. Systems that have $\mathbf{K = I}$ will be called orthonormal systems. Note

that there is a similar relationship between the **W** matrix and the $\mathbf{G}(s)$ functions. These correlation matrices will prove useful in finding an "orthonormal" ladder structure presented in chapter 3.

2.5. Summary

In this chapter, the notation usage and the concepts of expectation, cotrelation and norm were explained. Also, some adaptive filter theory was presented. In particular, the steepest descent LMS algorithm was described along with one benefit of orthonormal signals. As well, some difficulties associated with adapting IIR filters were described. Finally, some state-space background theory was presented including the definition of intermediate-functions and a transposed system where the intermediate-functions are exchanged. Sensitivity equations were also presented for the system coefficients. The state-space section ended with a definition of the correlation matrices **K** and **W.**

Chapter 3

Orthonormal Ladder Filters

3.1. Introduction

This chapter will present a new continuous-time state-space filter structure which has some interesting properties that are useful in the design of both adaptive and fixed filters. We call the filters with this new structure "orthonormal ladder filters." The name of this new structure is derived from two of the properties that are inherent to the structure. One property is that all the state signals are orthogonal when white noise is applied to the input of the filter with the norms of each of the state signals being equal. Such a property implies that the set of state signals are an *orthonormal* set. The other property inherent to this structure is the fact that the state signals are scaled versions of capacitor voltages and inductor currents of a singly-terminated *ladder* when the input is applied to the terminating resistor. The fact that the integrator outputs are orthogonal with a white noise input is useful when applied to an adaptive linear combiner, as was described in chapter 2. The close relationship to singly-terminated ladders allows a simple synthesis procedure and a trivial stability check. Recall from chapter 2 that a simple stability check is useful in adaptive IIR applications.

Although the material in this thesis is mainly concerned with the design of adaptive filters, orthonormal ladder filters are also useful in the design of fixed filters. For this reason, much of the material presented in this chapter will focus on the design aspects of the structure to a known transfer function.

One of the more interesting properties of orthonormal ladder filters is the fact that the resulting circuits are inherently scaled for optimum dynamic range. Moreover, an L_2 norm is used in dynamic range scaling as opposed to an L_∞ norm. An L_2 norm is equivalent to the norm for finite energy signals defined in chapter 2. Simply stated, L_2 scaling implies that the output of each integrator will have the same RMS value when white noise is applied at the filter input. In contrast, L_∞ scaling ensures that all integrator outputs will obtain the same peak voltage when a swept sinusoid is applied at the filter input. The issue of the relative merits of L_2 and L_∞ scalings is controversial. L_∞ is often used in analog systems while L_2 scaling is widely used in digital systems. L_2 is more realistic in many applications in the sense that it deals with inputs having a broadband spectrum (eg. speech) rather than sinusoids. However, L_2 scaling is less conservative in that it could cause clipping with sine-wave inputs in high-Q cases. In spite of this fact, it is felt that L_2 scaling covers a more general class of filters than L_∞. Note that while L_2 scaling is relatively difficult to apply to a cascade of biquads, the actual structure of orthonormal ladder filters ensures optimum dynamic range scaling with an L_2 norm.

Another useful property of orthonormal ladder filters is the ability to realize any stable transfer-function. Arbitrary poles are realized using the ladder feedback structure while transmission zeros are realized using an output summing stage. While output summing is often avoided in practice because of fears of poor stopband sensitivity properties, it will be shown that an orthonormal ladder filter (including, of course, the output summing stage) has a sensitivity performance comparable to a good design based on cascading biquads. Additionally, since for a given transfer-function the orthonormal ladder realization is unique, the design procedure is more easily automated than the process of finding an optimal biquad cascade design where pole-zero pairing and cascade ordering are important [Sedra and Brackett, 1978]. As well, in implementations where an output summing stage is difficult to realize, it will be shown that the

output summing stage can be replaced by using feed-forward to each of the inputs of the integrators.

State-space orthonormal IIR filter structures are well known in the digital filter literature [chapter 10, Roberts and Mullis, 1987]. One of the reasons for their use is that overflow oscillations are impossible in these digital filters. However, their main disadvantage is that their structure is fairly dense. Fortunately, the structure for continuous-time orthonormal ladder filters is quite sparse.

For an orthonormal filter, the state correlation matrix, K, is the identity matrix. Therefore, a simple formula is required relating the state correlation matrix and state-space system matrices in order to find orthonormal systems. We derive such a formula in section 3.2. Although this formula is well known in the control literature, it is derived here to emphasize its physical interpretation. Though there are many structures for orthonormal systems, this chapter deals with one in particular, the orthonormal ladder filter described in section 3.3. In this section the synthesis procedure for orthonormal ladder filters is described using the relationship of the structure to singly-terminated LC ladders. In section 3.4, a very simple stability test for orthonormal ladder filters is presented and in section 3.5, an example of an orthonormal ladder design is given. Finally, section 3.6 presents a sensitivity and dynamic range comparison between designs based on orthonormal ladder filters, operational simulations of doubly-terminated LC ladders, and cascades of biquads.

3.2. State-correlation matrices and the Lyapunov equation

We have, from chapter 2, that the state-correlation matrix K is defined as

$$K_{ij} = \langle F_i(s), F_j(s) \rangle = \int_{-\infty}^{\infty} F_i(j\omega)\bar{F}_j(j\omega)d\omega \tag{3.1}$$

which, by Parseval's relation, equals the inner product in the time domain given by

$$K_{ij} = \langle f_i(t), f_j(t) \rangle = 2\pi \int_{0}^{\infty} f_i(t) f_j(t) dt \tag{3.2}$$

The vector of functions $\mathbf{F}(s)$ is defined as

$$\mathbf{F}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} \tag{3.3}$$

and taking the inverse Laplace transform of $\mathbf{F}(s)$ to obtain a vector of impulse responses, $\mathbf{f(t)}$, one obtains

$$\mathbf{f(f)} = e^{\mathbf{A}t}\mathbf{b} \tag{3.4}$$

Substituting equation (3.4) into (3.2), we can write the matrix $\mathbf{K}$ as

$$\mathbf{K} = 2\pi \int_{0}^{\infty} e^{\mathbf{A}t}\mathbf{b}\mathbf{b}^T e^{\mathbf{A}^T t} dt \tag{3.5}$$

Differentiating the integrand, we find

$$\frac{d(e^{\mathbf{A}t}\mathbf{b}\mathbf{b}^T e^{\mathbf{A}^T t})}{dt} = \mathbf{A}e^{\mathbf{A}t}\mathbf{b}\mathbf{b}^T e^{\mathbf{A}^T t} + e^{\mathbf{A}t}\mathbf{b}\mathbf{b}^T e^{\mathbf{A}^T t}\mathbf{A}^T \tag{3.6}$$

Integrating both sides of equation (3.6) from 0 to ∞, results in

$$(e^{\mathbf{A}t}\mathbf{b}\mathbf{b}^T e^{\mathbf{A}^T t}) \Big|_0^{\infty} = \mathbf{A}\left[\int_0^{\infty} e^{\mathbf{A}t}\mathbf{b}\mathbf{b}^T e^{\mathbf{A}^T t} dt\right] + \left[\int_0^{\infty} e^{\mathbf{A}t}\mathbf{b}\mathbf{b}^T e^{\mathbf{A}^T t} dt\right]\mathbf{A}^T \tag{3.7}$$

Assuming the $\mathbf{A}$ matrix results in stable systems, the left side becomes $-\mathbf{b}\mathbf{b}^T$ while the matrix $\mathbf{K}$ can be substituted into the right side. This leads to the following Lyapunov equation.

$$\mathbf{A}\mathbf{K} + \mathbf{K}\mathbf{A}^T + 2\pi\mathbf{b}\mathbf{b}^T = 0 \tag{3.8}$$

This equation allows one to find the correlation matrix, $\mathbf{K}$, given the system matrices, $\mathbf{A}$ and $\mathbf{b}$. Note that the correlation matrix, $\mathbf{K}$, is called the controllability grammian in the control

literature [Brockett, 1970][Chen, 1984] and that a similar equation is obtained in the discrete-time domain [Roberts and Mullis, 1987].

It should be pointed out that the Lyapunov equation (3.8) above can also be used to test the stability of the **A** matrix. From the control literature, it can be shown that the matrix **A** is stable if the matrix **K** is positive definite and the positive semidefinite square root of $\mathbf{bb}^T$ together with **A** is observable [Wonham, 1985]. Since in orthonormal systems, **K** is the identity matrix (**I** is positive definite), one need only check that the observable constraint is satisfied to determine the stability of **A.** Although this fact is not explicitly used in this thesis, it could be used to check the stability (and hence usefulness) of orthonormal structures other than the one described in this chapter.

Before leaving this section, it should be noted that a similar relation can be found between the **W** correlation matrix and **A** and c. This relationship is

$$\mathbf{A}^T\mathbf{W} + \mathbf{W}\mathbf{A} + 2\pi\mathbf{c}\mathbf{c}^T = \mathbf{0} \tag{3.9}$$

3.3. Orthonormal ladder filter synthesis

As previously mentioned, orthonormal systems are obtained when **K** is the identity matrix. Therefore, substituting in **I** for **K** in equation (3.8), we have the following equation that must be satisfied for an orthonormal system.

$$\mathbf{A} + \mathbf{A}^T = -2\pi\mathbf{b}\mathbf{b}^T \tag{3.10}$$

Consider the state-space structure whose **A** and **b** matrices are given by

$$\mathbf{A} = \begin{bmatrix} 0 & \alpha_1 & & & 0 \\ -\alpha_1 & 0 & \alpha_2 & & \\ & -\alpha_2 & 0 & & \\ & & \ddots & \ddots & \alpha_{N-1} \\ & & & -\alpha_{N-1} & -\alpha_N \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \left(\frac{\alpha_N}{\pi}\right)^{\frac{1}{2}} \end{bmatrix} \tag{3.11}$$

where all α_i's are greater than zero. The **A matrix** is tridiagonal and is very nearly skew-symmetric except for the single non-zero diagonal element. The **b** vector consists of all zeros except for the N'th element. This system satisfies equation (3.10) and is therefore an orthonormal system regardless of the actual element values. We call the above system an orthonormal *ladder* system for reasons which will become apparent shortly.

For the above system to be a useful design structure, a procedure is required to place the eigenvalues of **A,** or equivalently the poles of the system, at positions in the left-half plane dictated by the filter transfer-function to be realized. With this goal in mind, we find that the above structure is very similar to that of the state-space description of a singly-terminated LC ladder filter where the states are defined to be the inductor currents and capacitor voltages. For the even order case, the singly-terminated ladder is shown in figure 3.1 below. Here, the resistor value is defined to be 1 Ω without any loss of generality. As well, reactive components are labelled r_i where r_i is either the capacitor or inductor value. The matrices **A** and **b** of the state-space description of the ladder in figure 3.1 are found to be

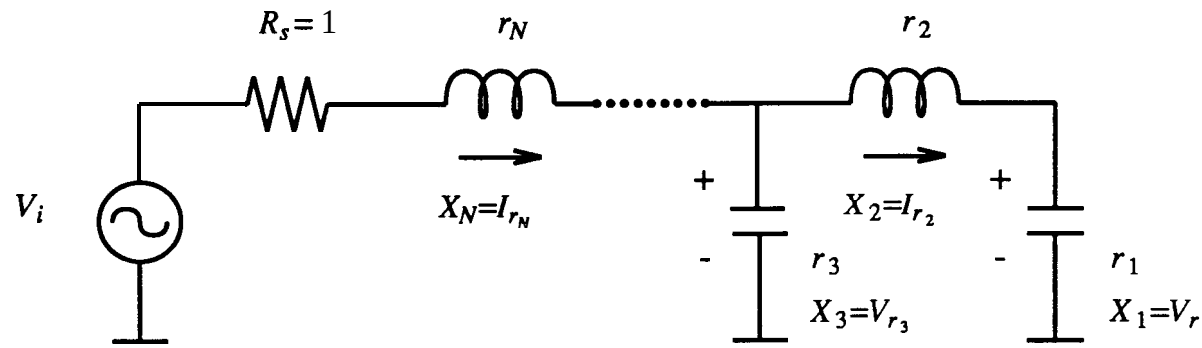


Figure 3.1: A singly terminated ladder and its states

$$\mathbf{A} = \begin{vmatrix} 0 & \frac{1}{r_1} & & & \mathbf{0} \\ -1 & 0 & \frac{1}{r_2} & & \\ r_2 & \frac{-1}{r_3} & 0 & & \\ & & & \ddots & \\ & & & 0 & \frac{1}{r_{N-1}} \\ 0 & & & -1 & -1 \\ & & & r_N & r_N \end{vmatrix} \quad \mathbf{b} = \begin{vmatrix} 0 \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ 1 \\ r_N \end{vmatrix} \tag{3.12}$$

We can transform the above system to that of the orthonormal ladder system with the structure of equation (3.11) by an appropriate scaling of the system states. Scaling the i'th state of a system by a factor β_i results in the i'th row of A and **b** being multiplied by β_i and the i'th column of A divided by β_i. Using this fact, the required scaling factors, β_i, are found to be

$$\beta_i = \left[\frac{r_i}{\pi}\right]^{\frac{1}{2}} \tag{3.13}$$

It should be noted that this scaling process does not change the system poles. Scaling the state-space description of equation (3.12) and comparing the result to the system in (3.11), we find the

following relationship between the elements of the orthonormal ladder system and the reactive components of the LC ladder:

$$\alpha_i = \left[\frac{1}{r_i r_{i+1}}\right]^{1} \qquad 1 \le i < N \tag{3.14}$$

$$\alpha_N = \frac{1}{r_N}$$

Recall that our goal is to be able to place the poles of the orthonormal ladder system at given locations in the left-half s-plane. This can be accomplished by obtaining a singly-terminated LC ladder with the desired poles and then using the above equation to obtain the elements of the orthonormal ladder system. From circuit theory, we know that any stable natural mode polynomial can be uniquely realized by an all-pole singly-terminated ladder with positive elements **[Humpherys, 1970].** Thus, one always finds a unique **A** matrix and **b** vector of an orthonormal ladder system for any set of stable poles.

Note that an interesting property of all-pole singly-terminated LC ladders has become apparent. We have shown that the states (inductor currents and capacitor voltages) of an all-pole singly-terminated LC ladder (when driven from the resistor) are all orthogonal since the states in equations (3.11) and (3.12) differ only in scaling. Also, the L_2 norms, of the ladder states are $\frac{1}{\beta_i}$ where β_i is given by equation (3.13). These simple properties appear to have never been mentioned in previous literature.

To implement the numerator of a particular transfer-function, the proper c vector must be obtained. To find the required c vector, we first need to find the states of the system. To find the states, note from figure 3.1 that the first state of the ladder, V_{r_1}, is an all-pole function with unity gain at DC. Hence, the numerator of the first state of the ladder is $E(0)$ where $E(s)$ is the natural

mode polynomial. Using this fact together with the state equations of the orthonormal ladder system, we can write the orthonormal states recursively as

$$F_1(s) = \frac{\beta_1 E(0)}{E(s)} \tag{3.15}$$

$$F_2(s) = \frac{s}{\alpha_1} F_1(s) \tag{3.16}$$

$$F_i(s) = \left[\frac{1}{\alpha_{i-1}}\right](sF_{i-1}(s) + \alpha_{i-2}F_{i-2}(s)) \qquad 3 \le i \le N \tag{3.17}$$

The proper c vector is found as the multiplying coefficients required to create the desired numerator.

We note from equations (3.15)- (3.17) that the numerators of the odd states will be even polynomials while the numerators of the even states will be odd polynomials. This fact helps to explain why an output summing amplifier implementing the c vector does not have poor sensitivity properties. Specifically, in the case of finite transmission zeros on the $j\omega$ axis, where the transmission-zero polynomial *P(s)* is purely even or odd, only even or odd elements of the c vector will be non-zero. Thus, a small change in any of the non-zero c elements will result in transmission zeros remaining on the $j\omega$ axis.

Figure 3.2 shows a block diagram of a general orthonormal ladder filter. The simple leap frog structure is a result of simulating a singly-terminated ladder. As shown in the block diagram, the output is obtained as a linear combination of the integrator outputs'.

Although output summing (having a c vector with more than one non-zero element) does not have poor sensitivity performance, there are situations where a circuit implementation of the c vector is difficult. An example of such a situation is the design of high frequency

[1] Note that, from equation (3.14), the units of α_i are *Hz as* expected. However, the units of the feed-in term are $\sqrt{Hz}$. This surd term is a result of forcing the states to have the same RMS value when a signal of constant spectral density in $V/\sqrt{Hz}$ is applied at the filter input,

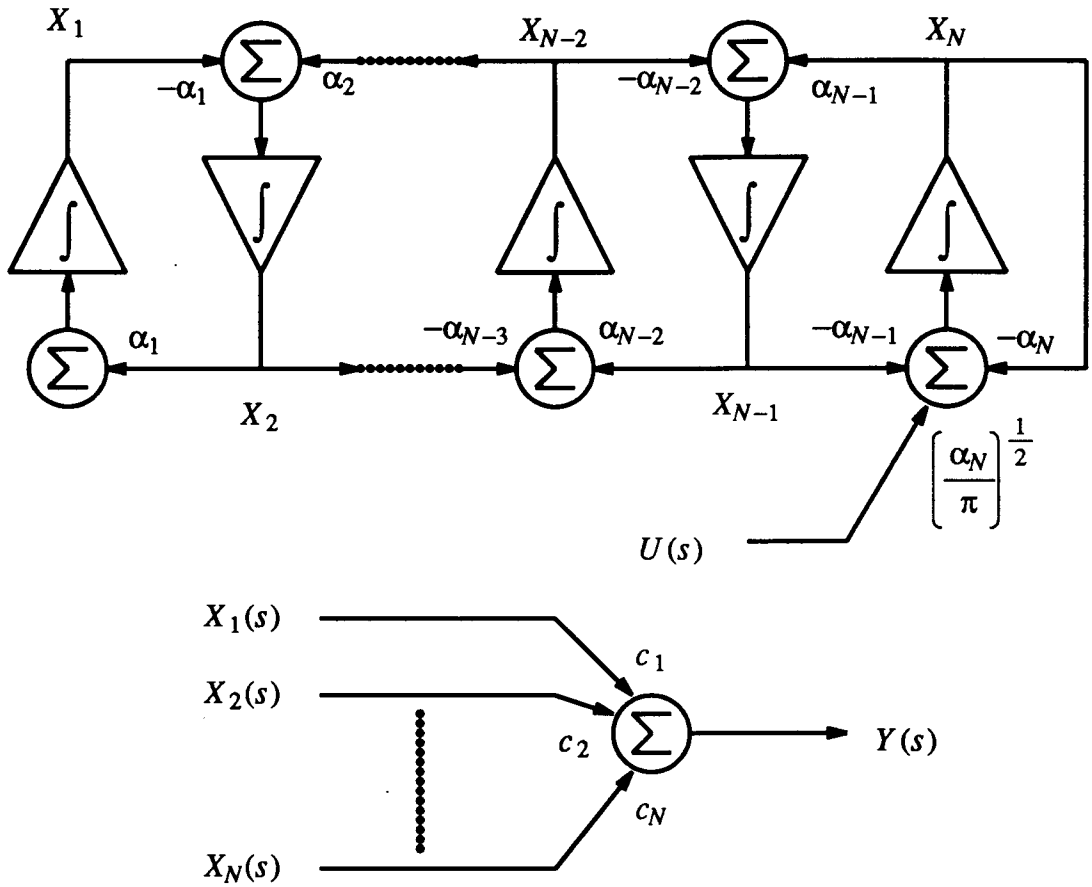


Figure 3.2: Block diagram of an orthonormal ladder filter

transconductance-C filters where a wide-band output summing network is difficult to implement. In such a situation, it is much easier to add one more input to each of the integrators than to design a high frequency summing stage with many inputs. For these situations, feed-forward (having a **b** vector with more than one non-zero element) can be used to create the required transmission zeros. It is important to note, however, that the feed-forward system to be described does not have an orthonormal set of **F** functions.

To create a feed-forward system, an orthonormal ladder system with output summing is first obtained. The feed-forward system can then be obtained as the transposed system of the orthonormal ladder system, as described in chapter 2. It is easily shown from equation (2.24) that the two systems have the same transfer-function. As well, the feed-forward system will have the same intermediate-transfer functions as the orthonormal ladder system but the two sets of functions will be exchanged. Thus, for the feed-forward system, the intermediate G functions are an orthonormal set. Since the intermediate-functions are simply interchanged, it is also easy to show from the sensitivity formulae in [Snelgrove and Sedra, 1986] that the feed-forward and orthonormal ladder systems will have the same sensitivity performance with respect to system elements. Finally, although the feed-forward system does not have the **F** functions scaled for optimum dynamic range, these functions can be L_2 **scaled** to equal levels using the standard method of scaling.

3.4. Stability test for orthonormal ladder filters

If orthonormal ladder filters are going to be used in actual adaptive IIR applications, it is often necessary to have a simple stability test for the **A** matrix. Since an orthonormal ladder filter simulates a passive **singly-terminated LC** ladder, a sufficient stability test is to check that all $\alpha_i, 1 \leq i \leq N$, are greater than zero. However, an even more trivial test can be derived.

Before developing this simple stability test, a comment should be made here about the situation where one of the $\alpha_i, 1 \leq i < N$ equals zero. This situation corresponds to a reactive element of the singly terminated ladder going to infinity. In this situation, part of the system will be decoupled from the damped portion of the system which may result in instabilities. For the stability test described below, we will make the assumption that none of the α_i coefficients equal zero.

Recall that scaling the states of a system does not change the system's poles. Therefore, we may scale any or all of the states by -1 and not affect the poles. Consider the case where the states 1 to j are scaled by -1 and $j<N$. In this case, the resulting system will be the same as the original system except that α_j is now less than zero. This scaling approach can be applied to obtain orthonorrnal ladder systems where the α_i's may be positive or negative except for α_N which will remain positive. Therefore, to test whether an orthonormal ladder system is stable, one need only check that α_N is positive and the other α_i are non-zero.

3.5. Design example

Consider the following fifth-order elliptic lowpass transfer-function with a 1 dB passband ripple,

$$T(s)=\frac{P(s)}{E(s)}=\frac{0.01321s^4+0.1037s^2+0.1739}{s^5+0.9287s^4+1.7726s^3+1.0557s^2+0.6917s+0.1739} \tag{3.18}$$

The reactive elements of the singly-terminated ladder realizing these poles can be found using continued fraction expansion [Humpherys, 1970] on the polynomial, $E(s)$. Applying such a procedure results in the following elements.

$$r_1=0.9078\,F \quad r_2=2.0205\,H \quad r_3=1.9937\,F \quad r_4=1.4606\,H \quad r_5=1.0768\,F \tag{3.19}$$

Using equation (3.14), the following elements of the orthonormal ladder system are obtained.

$$\alpha_1=0.7384 \quad \alpha_2=0.4982 \quad \alpha_3=0.5860 \quad \alpha_4=0.7934 \quad \alpha_5=0.9287 \tag{3.20}$$

The intermediate-functions of the orthonormal ladder system are found using equations (3.15)-(3.17) and are

$$F_1(s)=\frac{0.09346}{E(s)} \tag{3.21}$$

$$F_2(s)=\frac{0.1266s}{E(s)} \tag{3.22}$$

$$F_3(s) = \frac{0.2540s^2 + 0.1385}{E(s)} \tag{3.23}$$

$$F_4(s) = \frac{0.4335s^3 + 0.3440s}{E(s)} \tag{3.24}$$

$$F_5(s) = \frac{0.5437s^4 + 0.6181s^2 + 0.1018}{E(s)} \tag{3.25}$$

Finally, to obtain the c vector, we find the elements of c which satisfy the following equation.

$$[c_1F_1(s) + c_2F_2(s) + c_3F_3(s) + c_4F_4(s) + c_5F_5(s)]E(s) = P(s) \tag{3.26}$$

Solving for the c_i coefficients, we find the c vector and scalar d required to form the desired numerator to be

$$\mathbf{c}^T = [1.3163 \;\; 0 \;\; 0.3492 \;\; 0 \;\; 0.02431 \qquad d = 0 \tag{3.27}$$

3.6. Sensitivity performance comparison

This section will compare the sensitivity performance of orthonormal ladder filter realizations with realizations resulting from two alternate synthesis methods. One of the alternate methods *is* a state-space *simulation* of a doubly-terminated LC ladder filter [Johns et al, 1987]. The other method is a cascade of second-order sections implemented with Tow-Thomas biquads where the finite transmission zeros of the biquads are realized using feed-forward with a resistor and a capacitor [Sedra and Brackett, 1978]. Pole-zero pairing and cascade ordering are chosen using the rule-of-thumb in [Moschytz, 1975]. To use the analysis methods in ISnelgrove and Sedra, 1986], we require the cascade structure in a state-space formulation. Fortunately, a cascade of biquads design can be easily put into a state-space description if one allows a non-constant feedback matrix. The non-constant feedback matrix, A(s) consists of two matrices, $\mathbf{A}_1$ and $\mathbf{A}_2$, such that

$$\mathbf{A}(s) = \mathbf{A}_1 + s\mathbf{A}_2 \tag{3.28}$$

With an active-RC circuit, the $\mathbf{A}_2$ elements are realized with capacitor feed-ins to integrators. Finally, for a fair comparison with orthonormal ladder filters, L_2 dynamic range scaling was performed on all filters before comparing sensitivity or dynamic range.

Since different criteria are used to judge the filter performance in the passband and stopband, slightly different measures will be used in the two regions. However, in both bands, the multiparameter sensitivity measure presented by Schoeffler [Schoeffler, 1964] is used to find the standard deviation in the transfer-function for standard deviations of 1 percent of the nominal component values. The transfer-function deviation, $\sigma|T(j\omega)|$, is found from

$$\sigma|T(j\omega)| = 0.01\left|\sum_{=A_{ij},b_i,c_i,\gamma_i}\left[\frac{\partial|T(j\omega)|}{\partial x}x\right]^2\right|^{\frac{1}{2}} \tag{3.29}$$

where $\frac{\gamma_i}{s}$ represents the gain of the i'th integrator. Formulae in ISnelgrove and Sedra, 1986] were used to compute the derivative in equation (3.29). Changes in the elements of A_{ij}, b_i, and c_i directly correspond to changes in the feed-in resistors and capacitors of an active-RC implementation whereas changes in the elements γ_i correspond to changes in the integrating capacitors. Therefore, this deviation measure takes into account all the passive elements of an active-RC implementation.

Since transfer-function deviation is often the most critical performance measure in the passband, the passband deviation in dB, $D(\omega)$, is used to measure sensitivity performance in the passband. $D(\omega)$ is found from $\sigma|T(j\omega)|$ and $|T(j\omega)|$ as

$$D(\omega) = 20\log_{10}\left[1 + \frac{\sigma|T(j\omega)|}{|T(j\omega)|}\right] \tag{3.30}$$

This passband measure gives the standard deviation of the passband in dB from the ideal

response for standard deviations of 1 percent in component values.

In the stopband, an expected gain curve is plotted. This stopband expected gain value, $T_\sigma(\omega)$, is found from

$$T_\sigma(\omega) = 20\log_{10}(|T(j\omega)| + \sigma|T(j\omega)|) \tag{3.31}$$

This stopband performance measure allows one to easily see the expected stopband transmission for standard deviations of 1 percent of component values. Note that if the passband deviation measure were used in the stopband, it would go to infinity at transmission zeros.

For dynamic range comparisons, the figure of merit $\sum_i |G_i|_2^2$ will be used [Snelgrove and Sedra, 1986]. This figure of merit is the square of the rms noise level obtained when uncorrelated white noise sources of unit power spectral density are applied to each of the integrator inputs. Thus, a filter with good dynamic range will have a low number for $\sum_i |G_i|_2^2$.

For the fifth-order example above, three state-space descriptions were obtained using the different design approaches. The state-space description of the orthonormal system obtained for this example is

$$A = \begin{bmatrix} 0 & 0.7384 & 0 & 0 & 0 \\ -0.7384 & 0 & 0.4982 & 0 & 0 \\ 0 & -0.4982 & 0 & 0.5860 & 0 \\ 0 & 0 & -0.5860 & 0 & 0.7934 \\ 0 & 0 & 0 & -0.7934 & -0.9287 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.5437 \end{bmatrix} \tag{3.33}$$

$$c^T = [\ 1.3163\ 0\ \ 0.3492\ 0\ 0.0243\ ] \qquad d\ 0$$

The state-space system for the doubly-terminated ladder simulation is

$$\mathbf{A} = \begin{bmatrix} -0.4643 & -0.5823 & 0 & -0.0821 & -0.0045 \\ 0.8408 & 0 & -0.5994 & 0 & 0 \\ -0.1064 & 0.527\,1 & 0 & -0.4961 & -0.0272 \\ 0 & 0 & 0.6153 & 0 & -0.5892 \\ -0.0097 & 0.0479 & 0 & 0.7574 & -0.4643 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0.4655 \\ 0 \\ 0.1066 \\ 0 \\ 0.0097 \end{bmatrix} \tag{3.35}$$

$$\mathbf{c}^T = [\ 0\ 0\ 0\ 0\ 1.3620\ ] \qquad d = O$$

and the state-space system for the biquad cascade is

$$\mathbf{A} = \begin{bmatrix} -0.3379 & 0 & 0 & 0 & 0 \\ 0.7709 & 0 & -0.7967 & 0 & 0 \\ (0.0922)s & 0.6491 & -0.4577 & 0 & 0 \\ 0 & 0 & 1.4464 & 0 & -1.8603 \\ 0 & 0 & (0.3191)s & 0.5348 & -0.1330 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0.3297 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.37}$$

$$\mathbf{c}^T = [\ 0\ 0\ 0\ 0\ 1.3622\ ] \qquad d = O$$

Note that the first state of the biquad design is a first order lowpass function and this state is used as the input to the first biquad formed in states 2 and 3. The output of this first biquad is state 3 which is used as the input to the second biquad formed by states 4 and 5.

Figure 3.3 shows a plot of the ideal transfer-function response along with passband deviations, $D(\omega)$, and stopband expected gain, $T_\sigma(\omega)$ curves. We see from these curves that the orthonormal ladder system has a passband performance somewhere between the performance of the doubly-terminated ladder simulation and the biquad cascade. The stopband performance of the orthonormal ladder system is slightly worse than that of a cascade of biquads. The noise figures for the ladder, orthonormal, and cascade filters of this fifth-order example are 47, 65, and 117 respectively.

An eighth-order elliptic bandpass filter example presented in [Snelgrove and Sedra, 1986] was also investigated. For this eighth-order example, the resulting curves are shown in figure 3.4. We see from these curves that the orthonormal ladder filter still performs quite well in the passband and upper stopband but is slightly worse than the other two designs in the lower stopband. The reason for the poorer sensitivity performance at low frequencies and DC is explained as follows. The cascade design contains two bandpass filter biquads and therefore varying any of the components will not affect the two zeros at DC. Similarly, varying the component values

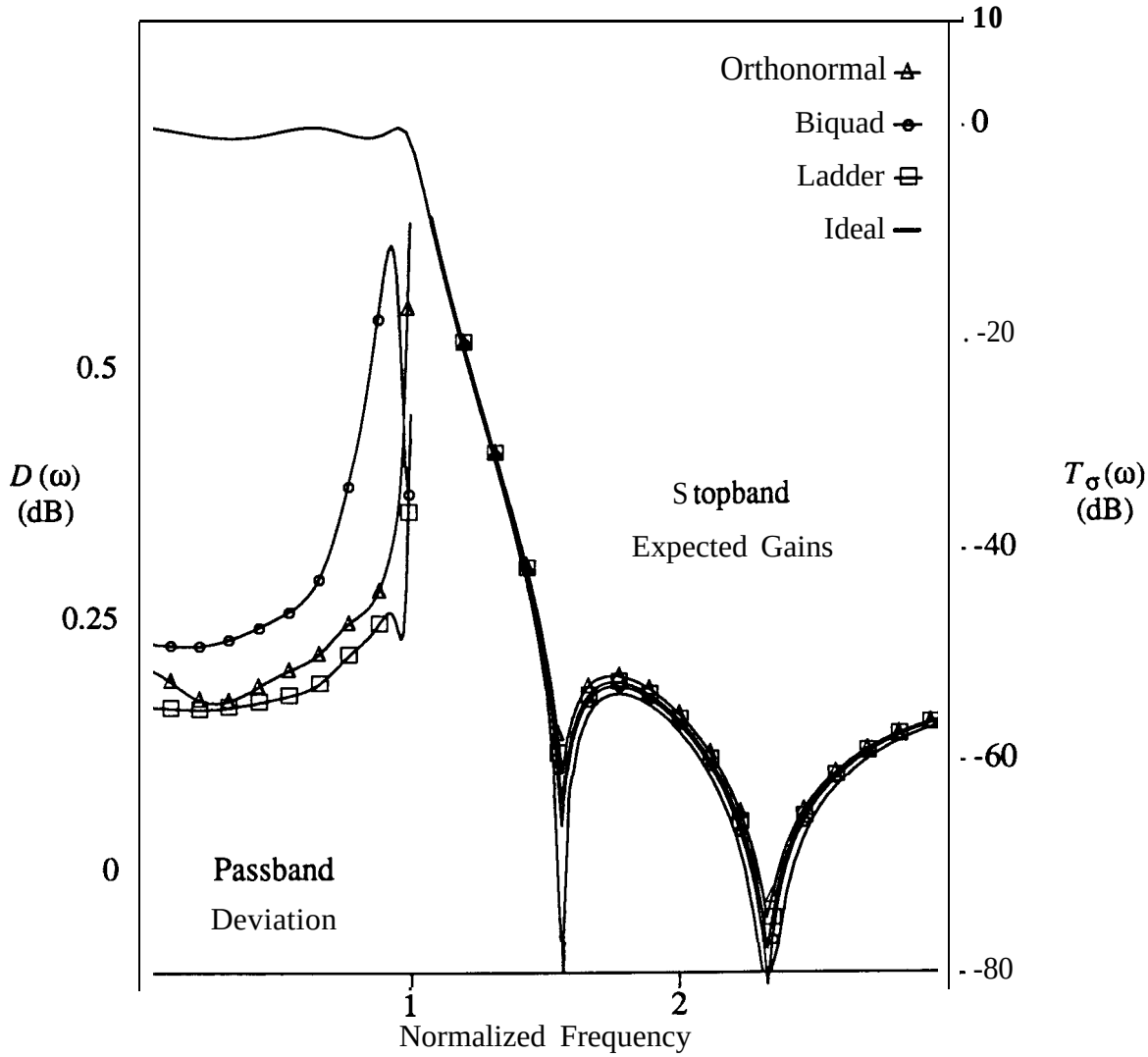


Figure 3.3: Fifth-order example: Plot of ideal transfer-function along with the expected stopband transmission, $T_\sigma(\omega)$, for a 1% component standard deviation. Also shown is the standard deviation in passband response, $D(\omega)$.

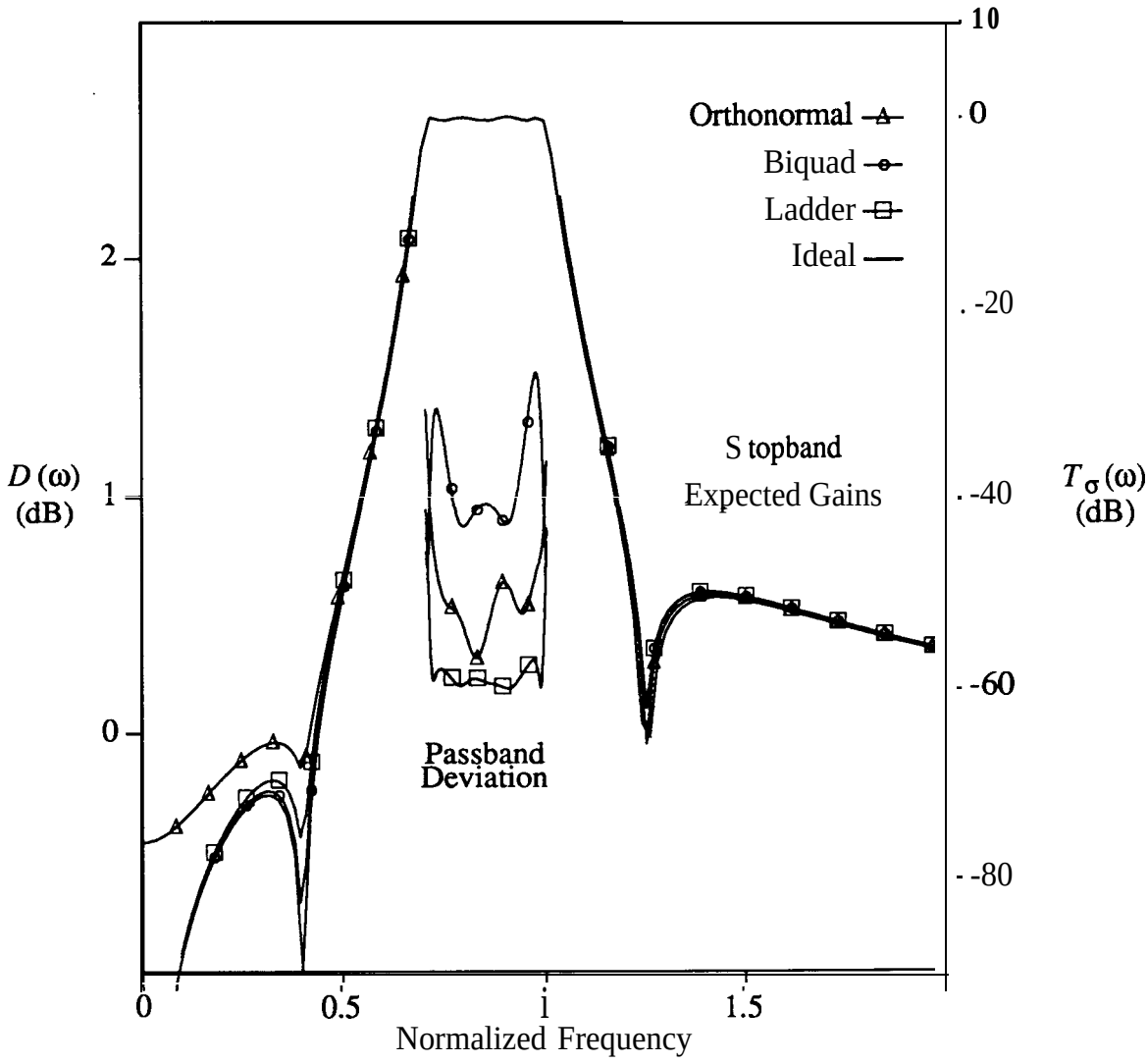


Figure 3.4: Eighth-order example: Plot of ideal transfer-function along with the expected stop-band transmission, $T_\sigma(\omega)$, for a 1% component standard deviation. Also shown is the standard deviation in passband response, $D(\omega)$.

of a doubly-terminated ladder having two zeros at DC will not affect the zeros. Since the ladder prototype has good sensitivity properties at DC, one would expect the active-RC simulation to also exhibit low sensitivities near DC and as seen from figure 3.4, this is the case. However, the orthonormal ladder filter creates the two zeros at DC by an output summing network and thus the

zeros will shift away from DC with component variations. The noise figures for the ladder, orthonormal, and cascade filter for this eighth-order example are 73, 100, and 15 1 respectively.

These two examples indicate that an orthonormal ladder filter has a passband sensitivity performance at least as good as a cascade of biquads (often much better) and a slightly worse stopband performance. The dynamic range performance of orthonoxmal ladder filters appears to fall between that obtained with LC ladder simulations and cascade designs.

3.7. Summary

A new filter structure called orthonormal ladder filters was presented. These filters are easy to synthesize through the use of singly-terminated LC ladders prototypes. As well, orthonormal ladder filters are automatically L_2 scaled for optimum dynamic range by the very nature of their structure. Also inherent in their structure is the fact that the integrator outputs are all orthogonal when the input is excited by white noise. We have also seen that orthonormal ladder filters can realize any stable transfer-function and have a performance comparable to a cascade of biquads. As well, it was shown that the sign of only one system coefficient determines the stability of the system.

It was also shown that a singly-terminated LC ladder driven through its terminating resistor has orthogonal states (inductor currents and capacitor voltages). As well, the L_2 norms of the ladder states were shown to have a simple relationship to the elements of the ladder. These simple relationships appear to have never been mentioned previously in the circuit theory literature.

Appendix 3.A

Laguerre Networks

Another structure that produces an orthonormal set of states when white noise is applied at the input is referred to as a Laguerre network. Although it is well known that Laguerre networks produce a set of orthonormal states [Lee, 1960], the mathematics to prove this property is fairly involved. This appendix will apply the Lyapunov equation derived in section 3.2 to prove this orthonormal property of Laguerre networks. Note that no claim is being made that this type of proof has not previously be presented, rather it is included in this chapter to show the generality of using the Lyapunov equation in finding orthonormal systems.

A Laguerre network consists of a first order lowpass filter followed by a cascade of first order all-pass functions as shown in figure 3.A. 1. Noting that the transfer function for $X_1(s)$ is

$$\frac{X_1(s)}{U(s)} = \frac{\sqrt{P/\pi}}{s+P} \tag{3.A. 1}$$

we can write the state equation for the state $X_1(s)$ as

$$sX_1(s) = -PX_1(s) + \sqrt{P/\pi}\,U(s) \tag{3.A.2}$$

The transfer function for $X_2(s)$ is seen to be

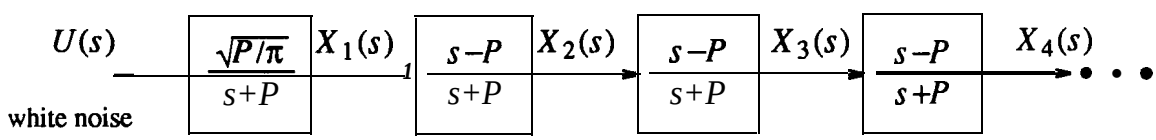


Figure 3.A.l: An orthonormal system resulting from Laguerre networks.

$$\frac{X_2(s)}{X_1(s)} \underline{S} \frac{-P}{s+P} \tag{3.A.3}$$

From this, we can write an equation

$$sX_2(s) = -PX_2(s) + (s-P)X_1(s) \tag{3.A.4}$$

and substituting in equation 3.A.-1 above to obtain a state equation, we have

$$sX_2(s) = -PX_2(s) - 2PX_1(s) + \sqrt{P/\pi}\,U(s) \tag{3.A.5}$$

Carrying on this procedure, we find the state equation for $X_3(s)$ to be

$$sX_3(s) = -PX_3(s) - 2PX_2(s) - 2PX_1(s) + \sqrt{P/\pi}\,U(s) \tag{3.A.6}$$

With this iterative procedure, it is not difficult to show that, in general, the state space description for the given Laguerre network has an **A** matrix which is lower triangular and all elements in the diagonal are -*P* and all elements below the diagonal are -2P. As well, the **b** vector has all elements equal to $\sqrt{P/\pi}$. In the case of a fourth order system, the following **A** and **b** coefficients are obtained:

$$\mathbf{A} = \begin{bmatrix} -P & 0 & 0 & 0 \\ -2P & -P & 0 & 0 \\ -2P & -2P & -P & 0 \\ -2P & -2P & -2P & -P \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \sqrt{P/\pi} \\ \sqrt{P/\pi} \\ \sqrt{P/\pi} \\ \sqrt{P/\pi} \end{bmatrix} \tag{3.A.7}$$

Since these coefficients satisfy the Lyapunov equation

$$\mathbf{AK} + \mathbf{KA}^T + 2\pi\mathbf{bb}^T = 0 \tag{3.A.8}$$

with **K** = I, the stated orthonormal property is true.

Chapter 4

Adaptive Recursive State-Space Filters

4.1. Introduction

This chapter will present new algorithms for adapting the poles and zeros of state-space filters. The adaptive algorithms presented in this chapter are described in the discrete time domain but these algorithms can be easily modified for the continuous time domain.

It should be pointed out that all the adaptive algorithms presented in this chapter are based on the LMS steepest descent approach and thus, depending on the performance surface, may converge to a local rather than global minimum. Although, this type of convergence may seriously limit the usefulness of this approach, there is an indication that if one increases the adaptive filter's order, only a global minimum will exist [Stearns, 1981]. As well, there presently exist adaptive algorithms which guarantee global convergence on direct form structures [Larimore et al, 1980] [Fan and Jenkins, 1986] and thus it may be possible to modify the algorithms presented in this chapter to also ensure global convergence.

The first algorithm presented is intended for a general state-space recursive filter. Having the ability to adapt arbitrary state-space filters gives the designer the freedom to explore the performance advantages of different structures. Unfortunately, the computation requirements to adapt a general state-space gradient filter is quite high. However, it will be shown that the amount of computation can be reduced by adapting any single column of the feedback matrix. Alternatively, in the special case of a single-element input summing vector and a small feedforward component from the input directly to the output, a single row of the feedback matrix can be

adapted with reduced computation. It will be shown that in applications where final pole locations can be estimated, these new adaptive filter structures have much faster adaptation rates than the traditional approach using direct-form filters. As well, the noise performance of these new structures will be shown to be significantly better than the direct form case.

Up to this point, we have been dealing with continuous time state-space systems whereas digital state-space systems are assumed in this chapter. Thus, we will begin by describing digital state-space systems in section 4.2. Also presented in this section are modified sensitivity formulae which can be used to adapt the filter coefficients. These gradient formulae arc used to find a minimum in the performance surface. In section 4.3, the adaptation algorithm for a general state-space filter is described. Unfortunately, for this general case, obtaining the gradients is computationally intensive. In section 4.4, a single-column adaptation algorithm is presented that has significantly less computations required to compute the gradients. As well, sufficiency tests are developed to help one check whether a column of a particular design can be adapted such that arbitrary pole locations may be obtained. Also presented in this section is a single-row adaptationstructure. A noise performance comparison between different filter structures is presented in Section 4.5 to illustrate the advantage of single column and single row adaptation over direct-form structures. To compare the different rates of adaptation, simulation results for a number of examples are given in Section 4.6.

4.2. Digital state-space systems

Similar to continuous time state-space systems, an N'th order state-space digital filter can be described by the following equations:

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{b}u(n) \tag{4.1}$$

$$y(n) = \mathbf{c}^T \mathbf{x}(n) + du(n)$$

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

$$\frac{Y(z)}{U(z)} = \mathbf{c}^T(z\mathbf{I}-\mathbf{A})^{-1}\mathbf{b} + d \qquad \textbf{(4.2)}$$

This transfer function equation is similar to the corresponding equation for the continuous time domain and therefore, as before, the poles of the system are determined by the **A** matrix (the poles are simply the eigenvalues of **A).**

As in the continuous time domain, two sets of intermediate-transfer functions, **F(z)** and G(z) can be defined. The first set, F(z), consists of the transfer functions from the filter input to the filter states.

$$\mathbf{F(z)} = (z\mathbf{I}-\mathbf{A})^{-1}\mathbf{b} \qquad (4.3)$$

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.

$$\mathbf{G}^T(z) = \mathbf{c}^T(z\mathbf{I}-\mathbf{A})^{-1} \qquad \textbf{(4.4)}$$

To obtain gradient signals, we use the sensitivity formulae in chapter 2 (with z substituted for s) to relate the derivatives of the output signal with respect to each of the system coefficients and the intermediate-transfer functions.

$$\frac{\partial Y(z)}{\partial A_{ij}} = G_i(z)X_j(z) \qquad \textbf{(4.5)}$$

$$\frac{\partial Y(z)}{\partial b_i} = G_i(z)U(z) \qquad \textbf{(4.6)}$$

$$\frac{\partial Y(z)}{\partial c_i} = X_i(z) \tag{4.7}$$

$$\frac{\partial Y(z)}{\partial d} = U(z) \tag{4.8}$$

From the above equations it is obvious that the gradient signals required to adapt the c vector elements are available as the output states, $\mathbf{x}(n)$, while 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-transfer functions from the input to the states equal to G(z) of the original system. Fortunately, we can obtain this new system as the transposed system of the original system as described in chapter 2. How this new system is applied will become apparent in the next section.

4.3. LMS adaptive algorithm for state-space filters

A block diagram of a state-space recursive adaptive filter is shown in figure 4.1 where the state-space coefficients now change with each timestep and hence are functions of the timestep "n". The state-space system is shown as two separate blocks corresponding to the state-space describing equations. Specifically, the feedback matrix, **A,** and input summing vector, **b,** implement the first equation of a state-space system and create the state signals, $\mathbf{x}(n)$, as the outputs of the first block. These state signals together with the system input, u, are weighted using the output summing vector, 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, $y(n)$.

Recall from chapter 2 that during adaptation, coefficients of the adaptive filter are changed to minimize the mean squared error signal, denoted as $E[e^2(n)]$. The LMS algorithm for updating any coefficient, p, of the adaptive filter is

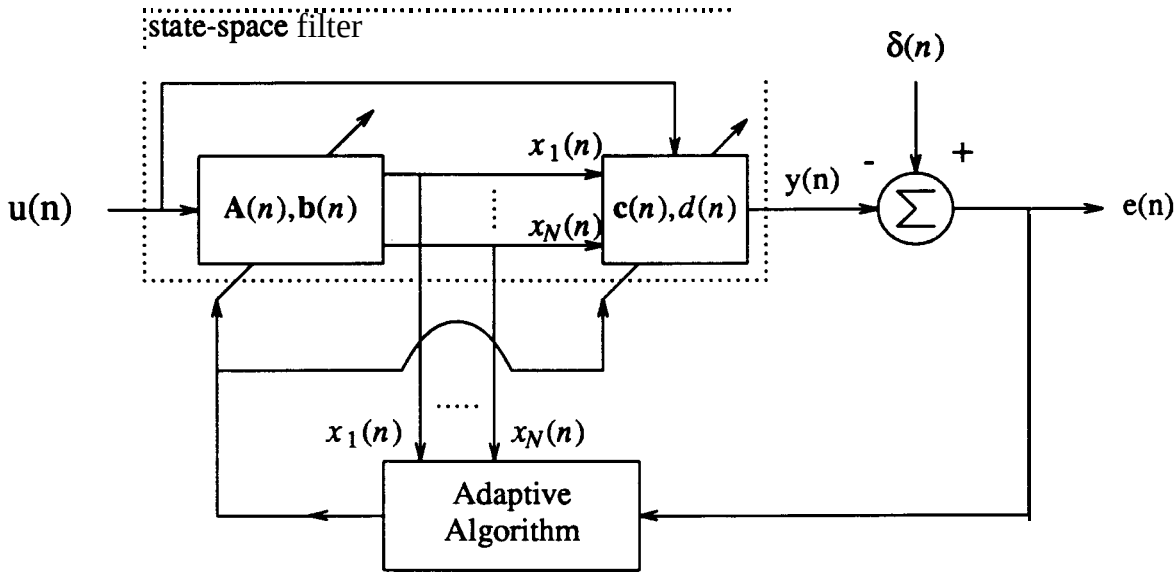


Figure 4.1: Adaptive state-space filter. The filter is shown in two separate blocks which correspond to the state-space describing equations.

$$p(n+1) = p(n) + 2\mu e(n)\frac{\partial y(n)}{\partial p(n)} \tag{4.9}$$

where μ is a step size to control convergence of the algorithm. We now assume we can write the following [Martin and Sun, 1986] [Yassa, 1987]

$$\frac{\partial y(n)}{\partial p} = Z^{-1}\left\{\frac{\partial Y(z)}{\partial p}\right\} \tag{4.10}$$

where Z^{-1} is the inverse z-transform. Substituting the gradient results of the previous section[1] in the update equation (4.9), the following adaptation equations for the system coefficients are obtained:

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

$$A_{ij}(n+1) = A_{ij}(n) + 2\mu e(n)\alpha_{ij}(n) \tag{4.11}$$

$$b_i(n+1) = b_i(n) + 2\mu e(n)\beta_i(n) \tag{4.12}$$

$$c_i(n+1) = c_i(n) + 2\mu e(n)x_i(n) \tag{4.13}$$

$$d(n+1) = d(n) + 2\mu e(n)u(n) \tag{4.14}$$

where

$$\alpha_{ij} = g_i(n) \otimes x_j(n) \tag{4.15}$$

$$\beta_i = g_i(n) \otimes u(n) \tag{4.16}$$

and the symbol $\otimes$ denotes convolution.

Note that the adaptation equations for the elements of A and **b** involve convolution while the elements of c and d have straightforward equations. **As discussed** previously, we can accomplish these convolutions by using the transposed system. **A** new system is created with the feedback matrix $\mathbf{A}^T$ and the input summing vector c. This new system has the impulse response $g_i(n)$ at the Output of the i'th state.

To implement the above adaptation equations, the filter structures shown in figure 4.2 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 **b** vector while each of the other transposed filters is used to adapt the elements of a column of the 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 the gradient equations obtained with this approach are identical to those obtained for direct form gradient adaptation in [Stearns et al, 1976][White, 1975]. However, note that an intermediate function approach was applied here whereas algebraic methods were used to obtain the gradient equations in the aforementioned references. This is to be expected since both methods are finding the derivative of the output with respect to the filter

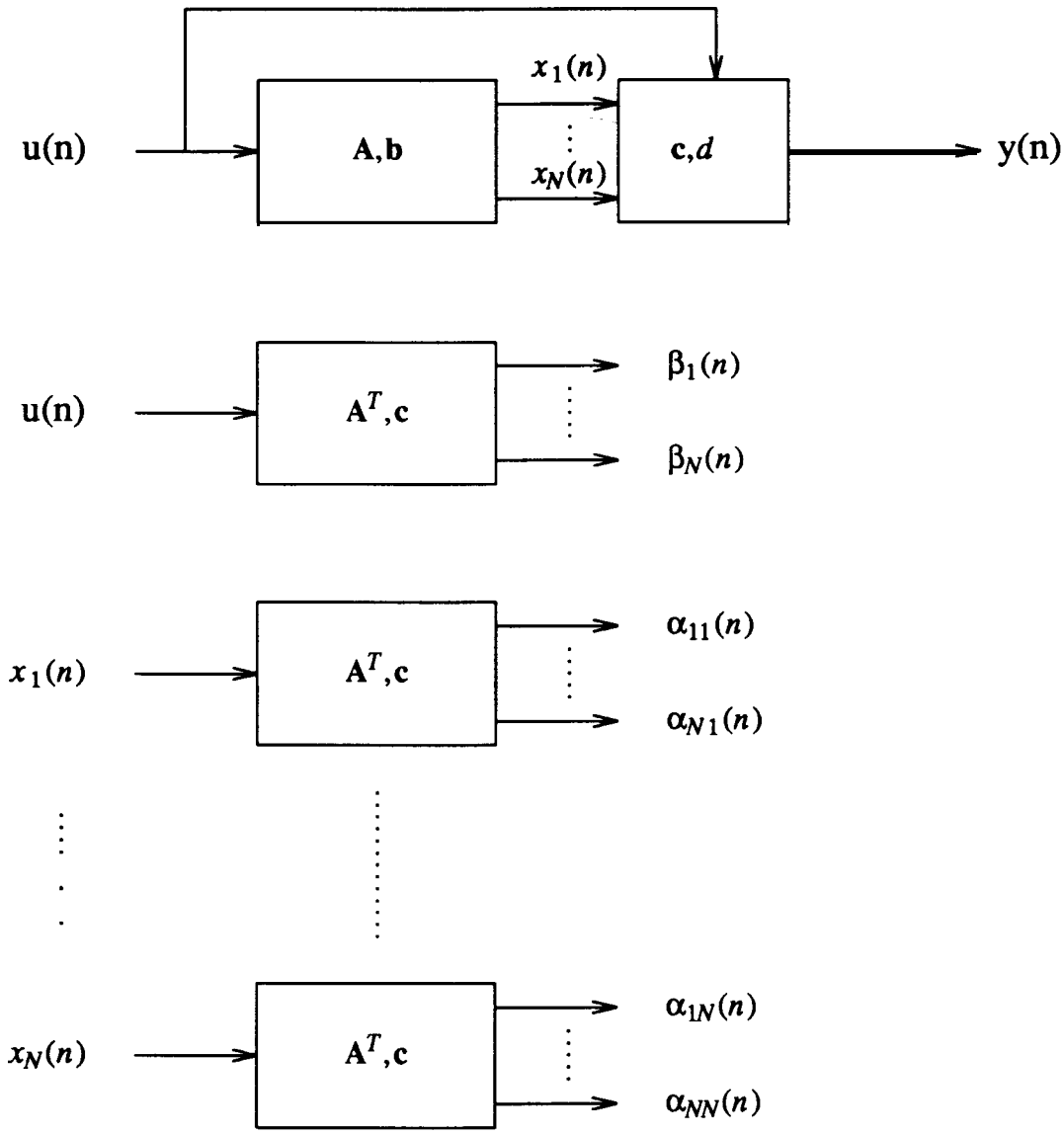


Figure 4.2: Generating the gradients for a general state-space adaptive filter

coefficients.

4.4. Reduced computation state-space adaptive filters

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

To further reduce the number of computations, 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 which can be adapted to any set of poles by changing only N elements of the A matrix.

4.4.1. Single column adaptive filters

Recall that each of the transposed filters of figure 4.2 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

$$\mathbf{A} = \begin{vmatrix} 0 & 0 & \cdot & 0 & 0 & a_1 \\ 1 & 0 & \cdot & 0 & 0 & a_2 \\ 0 & 1 & \cdot & 0 & 0 & a_3 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 1 & 0 & a_{N-1} \\ 0 & 0 & \cdot & 0 & 1 & a_N \end{vmatrix} \qquad (4.17)$$

However, we do not have to restrict ourselves to the modified direct form to obtain computational savings. From control theory, the pole assignment theorem [Wonham,1985] states the

following:

Pole Assignment Theorem

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

One way to look at this pole assignment theorem is to consider a state-space system with the fixed feedback matrix **A** and states x where one wishes to change the poles of the system to arbitrary locations by introducing extra feedback into the system. The extra feedback is introduced by taking a weighted sum of the states using the value of the **m** vector to create a feedback signal, y_f (equal to $\mathbf{m}^T\mathbf{x}$) and then applying the feedback signal, y_f, back into the states using the value of the vector, **k** (traditionally an input summing **b** vector). With this approach, it is not difficult to see that the poles of the new system are the eigenvalues of $\mathbf{A}+\mathbf{k}\mathbf{m}^T$. However, to ensure that arbitrary pole locations can be obtained, the feedback signal, y_f, must contain enough information about the states of the original system leading to the observability constraint in the theorem. The constraint is that the states of the original system must be observable through the signal y_f. Observability implies that with no input to the system, the initial states of the system can be determined by looking solely at the output signal. This observability constraint is also equivalent to having the intermediate G-functions of the system independent [Kuo,1980].

In the use of the pole assignment theorem in this thesis the **m** vector is restricted to be a basis vector $\mathbf{v}_i$ where a basis vector, $\mathbf{v}_i$, consists of all zeros except for the unit element in the i'th row. With this restriction and for a given **A** matrix, the pole assignment theorem states that we can obtain any desired set of poles by changing only the i'th column of **A** if $(\mathbf{v}_i^T, \mathbf{A})$ is observable (observability is further discussed 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 given in figure 4.3.

4.4.2. Sufficiency tests for column adaptation

As mentioned above in the pole assignment theorem, to adapt the i'th column of **A,** $(\mathbf{v}_i^T, \mathbf{A})$ must be observable. (Here, **m** has been replaced with $\mathbf{v}_i$ 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 which could be used. (For a discussion of observability tests, see [Kuo, 1980].) Unfortunately, filter designers do not always have access to software which can easily perform an observability test. For this reason, two simple sufficiency tests are presented here allowing a filter designer to check whether the above matrix pair satisfies the observability constraint. If the first test is satisfied then the particular column of **A cannot** be adapted while if the second test is satisfied then the column **can be** adapted. In most

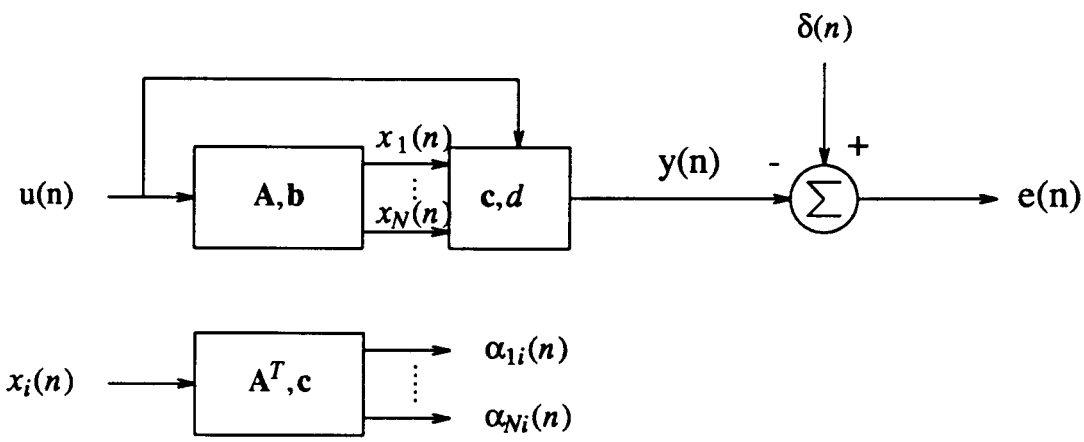


Figure 4.3: Generating the gradients for a single column adaptive filter

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 i'th column of **A to** obtain arbitrary pole locations. Define the vector of functions, $\mathbf{G}_{Oi}(z)$, as

$$\mathbf{G}_{Oi}^{T}(z) = \mathbf{v}_i^{T}(z\mathbf{I}-\mathbf{A})^{-1} \tag{4.18}$$

This vector of functions is most easily visualized as the intermediate G functions of the state-space system having a feedback matrix, **A,** and an output summing vector, $\mathbf{v}_i$; in other words, with the j'th element of $\mathbf{G}_{Oi}(z)$ being the transfer function from the input of the j'th delay operator to the i'th state. To derive the two sufficiency tests, an observability independence theorem is required which states the following [Kuo, 1980]: The pair $(\mathbf{v}_i^{T}, \mathbf{A})$ is observable if and only if the elements of $\mathbf{G}_{Oi}(z)$ are linearly independent (over the field of complex numbers).

The two sufficiency tests are:

Column Adaptation Test 1

If any of the elements of $\mathbf{G}_{Oi}^{T}(z)$ is zero then the i'th column of **A cannot** be adapted to arbitrary pole locations.

The proof for this test comes from the fact that the elements of $\mathbf{G}_{Oi}(z)$ are not independent if one of the elements is zero. Since the elements of $\mathbf{G}_{Oi}(z)$ are not independent then the observability independence theorem implies that the pair $(\mathbf{v}_i^{T}, \mathbf{A})$ is not observable. Therefore, by the pole assignment theorem, the i'th column of **A** cannot be adapted to realize arbitrary poles.

Column **Adaptation Test 2**

If any of the elements of $\mathbf{G}_{Oi}(z)$ is of order N then the i'th column of **A can** be adapted to realize arbitrary pole locations.

The proof for this test comes from the fact that the elements of $\mathbf{G}_{Oi}(z)$ are independent if one element is N'th order. This fact can be proved by contradiction. Assume a given system with

N delay elements where the elements of $\mathbf{G}_{Oi}(z)$ are dependent and one of the elements is N'th order. Since the elements of $\mathbf{G}_{Oi}(z)$ are dependent, at least one function can be created as a linear combination of the other functions. Therefore, a system with only N-l delay elements can be constructed with the same dependent set of $\mathbf{G}_{Oi}(z)$ elements. However, it is well known that an N'th order transfer function cannot be created from less than N delay elements. This contradicts the original assumption, therefore if one element is N'th order, the elements of $\mathbf{G}_{Oi}(z)$ must be independent. Since the elements of $\mathbf{G}_{Oi}(z)$ are independent, the observability independence and pole assignment theorems can be applied to prove the stated test.

As an example of these sufficiency tests, consider the fourth order **A** matrix

$$\mathbf{A} = \begin{bmatrix} 0 & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & 0 & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix} \tag{4.19}$$

This system is a cascade of two biquads where (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 figure 4.4).

Consider the case where one wishes to adapt the first column. It is clear from figure 4.4 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 $\mathbf{G}_{O1}(z)$ 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 as only feedforward terms are added from the first to the second biquad. A similar test shows that the second column cannot be adapted to realize arbitrary poles.

Now, consider the case of adapting the fourth column. In this case, the elements of $\mathbf{G}_{O4}(z)$ are the transfer functions from the inputs of the delay operators to the fourth state. Since the

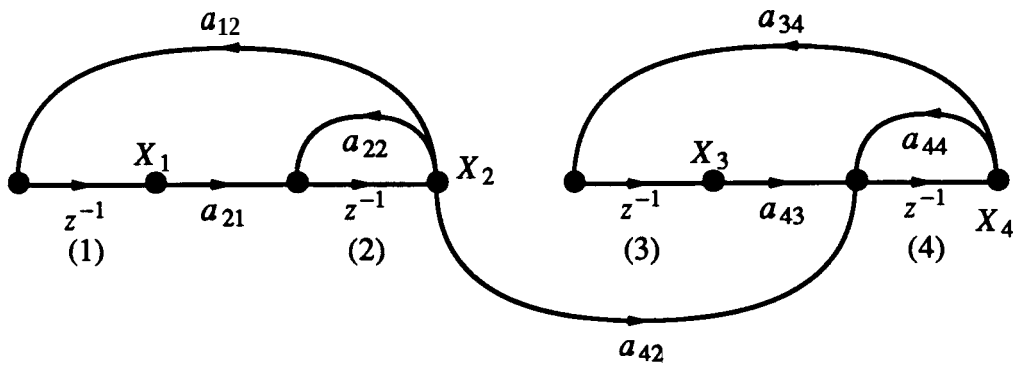


Figure 4.4: Signal flow graph for state-space system of cascade of two biquads. Only the A matrix branches are shown.

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 shows that the third column can also be adapted to realize arbitrary poles.

4.4.3. 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, $\mathbf{v}_i$ and the d coefficient 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 equation (4.11) for the i'th row, only one other system is required having 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 figure 4.5. Note that if the d element is close to zero then gradient signal obtained with this method will closely approximate the actual gradient signal.

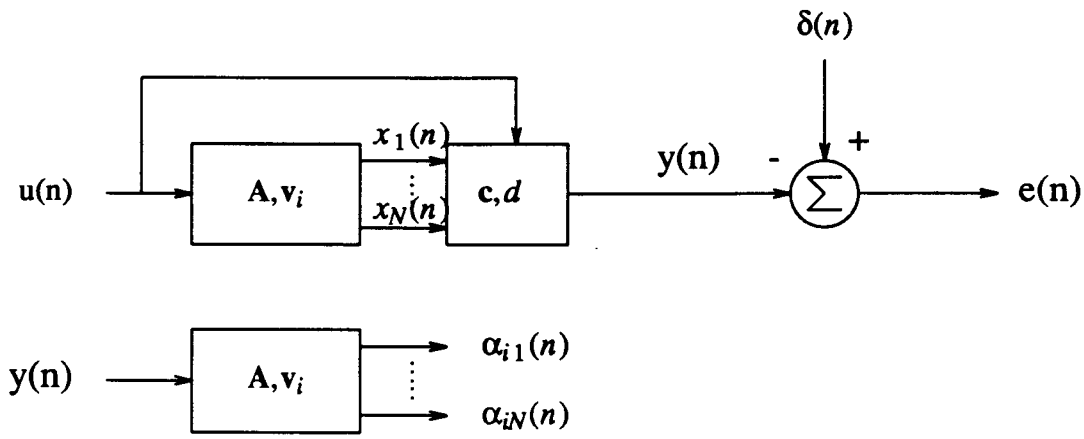


Figure 4.5: Generating the gradients for a single row adaptive filter

To determine whether the i'th row can be adapted to obtain arbitrary pole positions, one can use the controllability pole assignment theorem [Wonham, 1985]. As well, the sufficiency tests described above for adapting a column of the feedback matrix can be easily modified 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 [Yassa, 1987]. This method of obtaining gradients requires significantly less computations than that originally proposed in [Stearns et al, 1976][White, 1975]. Also note that the non-zero element of the input summing vector, **b**, does not have to equal one. If the non-zero element is not unity, then the above results still hold but the gradients for the i'th row will be scaled. Finally, note that the input vector must have only one non-zero element for efficient single row adaptation. This restriction is not present for single column adaptation.

4.5. 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 [Mullis and Roberts, 1976][Amit and Shaked, 1988],

$$N_M = trace(\mathbf{KWQ}) \tag{4.20}$$

where **K** and **W are the** state correlation matrices for digital state-space systems. In digital sys-terns, **K** and **W** satisfy the following equations [Mullis and Roberts, 1976]

$$\mathbf{K} = \mathbf{AKA}^T + \mathbf{bb}^T = \sum_{k=0}^{\infty} (\mathbf{A}^k\mathbf{b})(\mathbf{A}^k\mathbf{b})^T \tag{4.21}$$

$$\mathbf{W} = \mathbf{A}^T\mathbf{WA} + \mathbf{c}^T\mathbf{c} = \sum_{k=0}^{\infty} (\mathbf{cA}^k)^T(\mathbf{cA}^k) \tag{4.22}$$

The matrix Q is a diagonal matrix where Q_{ii} is zero if the elements in row i of **A** consist only of O's, 1 's and - 1 's. Otherwise, Q_{ii} is one.

Note that this noise measure is valid when using a modem digital signal processor having 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 non-zero element. The matrix Q makes an adjustment for rows where no truncation errors are introduced. In the case of a direct form filter, there are N-l rows where no truncation errors are present.

Note that the noise measure defined in [Mullis and Roberts, 1976] simply has Q equal to the identity matrix and thus using that measure would result in even higher noise figures for the direct-form case than those obtained in this thesis. Noise figures for other structures would

remain relatively unchanged.

With a noise measure to compare different filter structures, we may now proceed with an example. The example used will be a narrowband, 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 lowpass filter with a sampling frequency to passband frequency ratio of about 32. The same transfer function is used in the simulation results of section 4.6.2 where the poles and zeros of the transfer function are given (in the last row of table 4.1). Three different realizations of this filter are investigated with respect to noise performance.

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

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.8889 & -2.7432 & 2.8523 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.23}$$

$$\mathbf{c}^T = [\ 0.01003\ \ -0.01884\ \ 0.01088\ ] \qquad d = 0.005312$$

Using equations (4.21) and (4.22), the K and W matrices for the direct form realization are found to be

$$\mathbf{K} = \begin{bmatrix} 13395 & 13522 & 13395 \\ 13017 & 13395 & 13522 \\ 13522 & 13395 & 13017 \end{bmatrix} \tag{4.24}$$

$$\mathbf{W} = \begin{bmatrix} 0.0382 & -0.0800 & 0.0426 \\ -0.0800 & 0.1679 & -0.0898 \\ 0.0426 & -0.0898 & 0.0483 \end{bmatrix} \tag{4.25}$$

As well, for the state-space system shown in equation (4.23), by definition of the Q matrix, Q is seen to be

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (4.26)$$

Using equation (4.20), one can easily calculate the noise measure, N_M, for this filter to be 652. Thus, this noise measure is the value one would obtain in the case of an adaptive filter application using a direct form structure and the given final transfer function.

The next two filter realizations are obtained using a variation of the orthonormal filter structure described in chapter 3. Since the orthonormal filter structure gives good results in the continuous-time domain, it was felt that good filter performance would be obtained in a digital filter with a high ratio of sampling frequency to passband edge. (For design details, see appendix 4.A.) In this thesis, we shall refer to realizations obtained with this approach as "quasi-orthonormal filters" since the resulting realizations approach true orthonormal filters as the sampling frequency to passband edge is increased. While quasi-orthonormal filters are used in this section for comparison, the author believes that any good state-space design techniques [chapter 9, Roberts and Mullis, 1987] should give similar results.

Implementing the narrowband transfer function by a quasi-orthonormal filter, the following state-space system is obtained.

$$A = \begin{bmatrix} 1 & 0.1188 & 0 \\ -0.1188 & 1 & 0.1567 \\ 0 & -0.1567 & 0.8523 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0.2168 \end{bmatrix} \qquad (4.27)$$

$$\mathbf{c}^T = [\ 0.4755 \quad 0.0859 \quad 0.2168\ ] \qquad d = 0.005312$$

The K and W matrices for this structure are

$$K = \begin{bmatrix} 0.2202 & -0.01746 & -0.05179 \\ -0.01746 & 0.2941 & -0.04854 \\ -0.05179 & -0.04854 & 0.2456 \end{bmatrix} \qquad (4.28)$$

$$\mathbf{W} = \begin{bmatrix} 1.6698 & 1.0369 & 0.4573 \\ 1.0369 & 1.4300 & 0.9107 \\ 0.4573 & 0.9107 & 1.02656 \end{bmatrix} \tag{4.29}$$

Note that the **K** matrix indicates the structure is close to being orthonormal. Using the fact that the **Q** matrix for this structure is the identity matrix, the noise measure, N_M, is found to be 1.04. This noise figure is not much worse than the Mullis and Robert's optimal filter [Mullis and Roberts, 1976] which has a noise measure of 0.73 (the noise measure for the optimal filter can be calculated from the eigenvalues of **KW**). Thus, the quasi-orthonormal design approach appears to be a good structure for oversampled filters. Unfortunately, to obtain this low noise figure in an adaptive filter application, one would have to adapt the quasi-orthonormal structure where varying elements of the **A** matrix are in each of the rows and columns of **A**. Thus, a high computational load would be required to obtain the gradients for the elements of **A**.

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 above, 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. As well, the **c** vector and d scalar were changed to obtain the desired zeros. The following state-space system was obtained.

$$\mathbf{A} = \begin{bmatrix} 1 & 0.0577 & 0 \\ -0.0577 & 1 & 0.1633 \\ -0.1686 & -0.2163 & 0.8524 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0.309 \end{bmatrix} \tag{4.30}$$

$$\mathbf{c}^T = [\, 0.6984 \quad 0.0579 \quad 0.0352 \,] \qquad d = 0.00531$$

Finally, the following **K** and **W** were found for the above system.

$$\mathbf{K} = \begin{bmatrix} 0.1149 & -0.0187 & -0.1846 \\ -0.0187 & 0.6493 & -0.0814 \\ -0.1846 & -0.0814 & 0.7709 \end{bmatrix} \tag{4.31}$$

$$\mathbf{W} = \begin{bmatrix} 4.8721 & 1.4571 & 1.0212 \\ 1.4571 & 0.6479 & 0.4300 \\ 1.0212 & 0.4300 & 0.5054 \end{bmatrix} \tag{4.32}$$

The noise measure, N_M, for this filter is 1.4 which is slightly worse than the above 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.

4.6. Simulation results

This section will present computer simulation results of the above adaptive algorithms. The simulations are based on system identification applications where the adaptive filter is required to adjust its coefficients to match a reference transfer function. A block diagram of the application used for these simulations is shown in figure 4.6. The examples presented are grouped into two sets. The first group is intended to demonstrate that our state-space algorithm results in adaptation paths that closely follow those of steepest descent. The second group of examples indicates that much better adaptation rates may be obtained, in oversampled applications, by using structures other than direct-form.

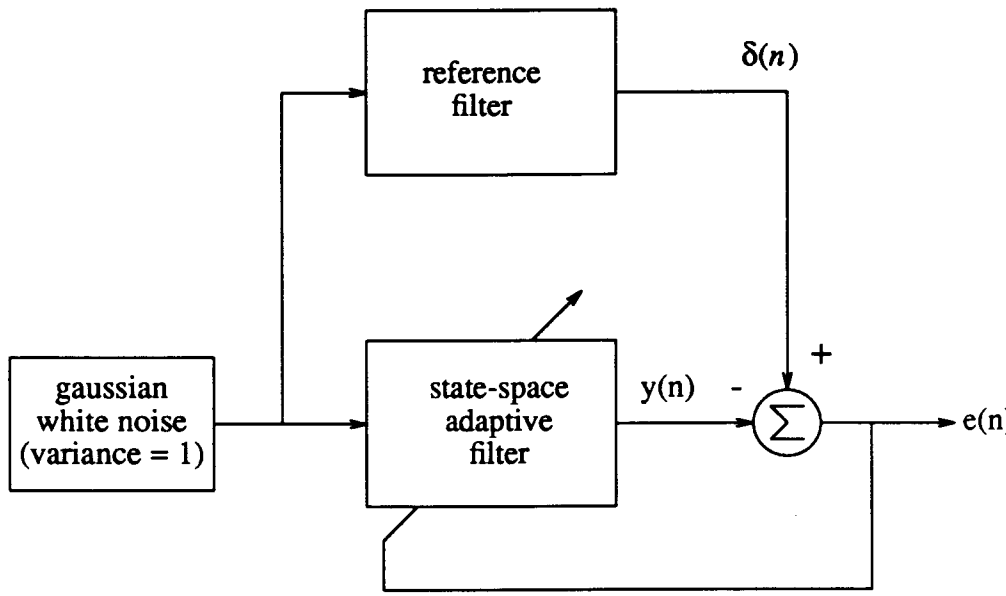


Figure 4.6: System identification application used for simulations

4.6.1. Adaptation paths

In the adaptation path examples, we shall use second-order reference transfer functions so that a contour plot of the error performance surface can be superimposed with the adaptation paths taken. For comparison purposes, we shall also plot the adaptation path taken by the approximate gradient algorithm proposed in [Feintuch, 1976]. (See appendix 4.B for an interpretation of the approximate algorithm in relation to the algorithm presented in this chapter).

The adaptive filters are implemented using the following direct form state-space system

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad (4.33)$$

$$\mathbf{c}^T = [\,0 \quad 1\,] \quad d = 0$$

For the state-space gradient approach, the single-row adaptation algorithm described in section 4.3 was used. For the approximate gradient approach, the algorithm described in appendix 4.B was used. For both algorithms, a small step size was used to observe the adaptation path of the coefficients.

For the first example, the following reference transfer function was used.

$$\frac{z^{-1}}{1 - 1.2z^{-1} + 0.7z^{-2}} \tag{4.34}$$

Therefore, after adaptation, the coefficients (a_1, a_2) should be (–0.7, 1.2). The starting point for both algorithms was at (–0.36, 0). Figure 4.7, below, shows the adaptation path of the coefficients superimposed on a contour plot of the error performance surface. In this case, the state-space gradient algorithm follows the steepest descent path, as expected, while the approximate gradient path is not far from the steepest descent path.

In the second example, the poles of the reference filter were chosen to be close to the unit circle with the location of the starting point poles even closer. This example is used to demonstrate that the state-space gradient method again follows the path of steepest descent while the approximate gradient approach deviates drastically. The reference transfer function used was

$$\frac{z^{-1}}{1 - 1.7z^{-1} + 0.8z^{-2}} \tag{4.35}$$

The starting point for the coefficients (a_1, a_2) was (–0.915, 1.7). The adaptation paths and contour plot for this example are shown in figure 4.8, below. From figure 4.8, we see that the state-space gradient algorithm still follows the path of steepest descent. However, the approximate gradient approach is quite far from the steepest descent path during the first half of adaptation and then follows the steepest descent path during the second half of adaptation.

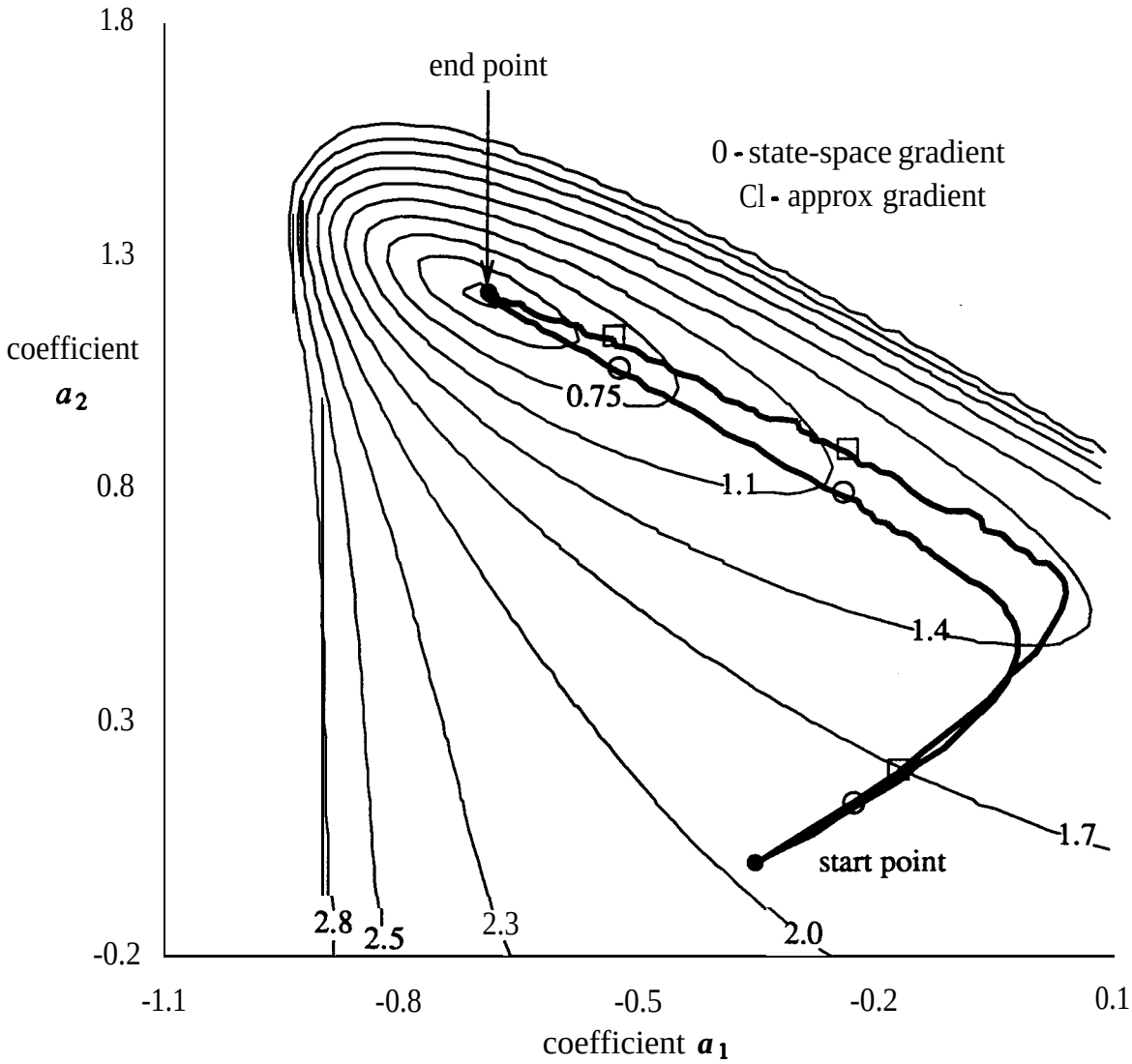


Figure 4.7: Example 1 of adaptation paths for state-space gradient method and method in [Feintuch, 1976] superimposed on contour plot of error performance surface.
start:$(a_1 = -0.36, a_2 = 0.0)$, end:$(a_1 = -0.7, a_2 = 1.2)$.

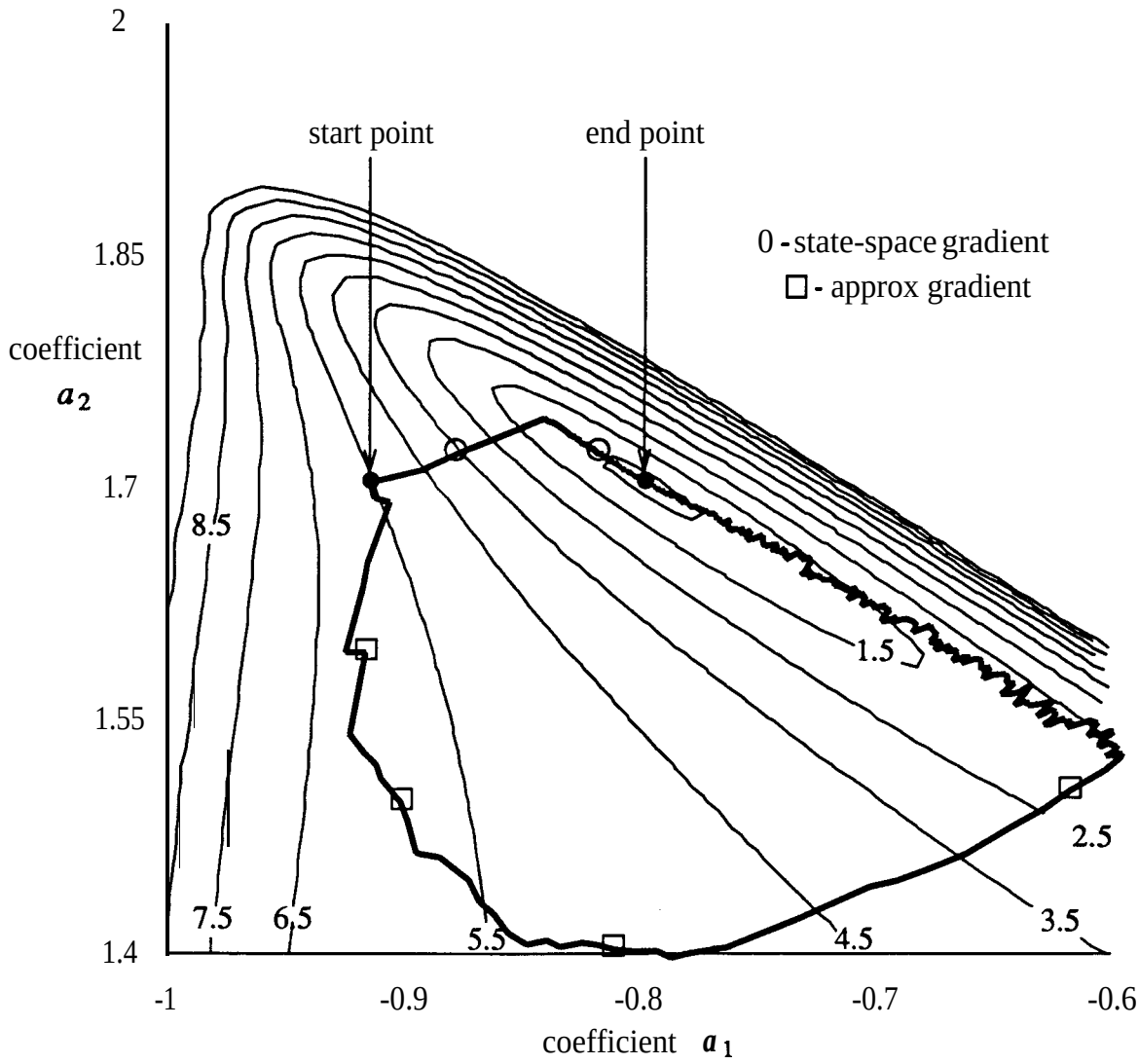


Figure 4.8: Example 2 of adaptation paths for state-space gradient method and method in [Feintuch, 1976] superimposed on contour plot of error performance surface. start:$(a_1 = -0.915, a_2 = 1.7)$, end:$(a_1 = -0.8, a_2 = 1.7)$.

The above simulations indicate that the state-space gradient algorithm should always converge to a minimum in the error performance surface while the performance of the method in [Feintuch, 1976] appears to be quite application dependent with no guarantee of converging to a minimum.

4.6.2. Narrowband examples

This section will present adaptation rate results for filters with varying ratios of sampling frequency to passband edge frequency. All the reference filters are derived from a third-order elliptic lowpass analog prototype with the following s-plane poles and zeros.

$$poles = \{ -0.3226, -0.1343 \pm j\, 0.9\,1920 \} \tag{4.36}$$

$$\text{zeros} = \{ \pm j\, 2.2705, \infty \}$$

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

To obtain narrowband digital filters with varying bandwidths, the bilinear transform [Oppenheim and Schafer, 1975] was applied to the analog prototype,

$$z = \frac{1 + (T/2)s}{1 - (T/2)s} \tag{4.37}$$

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 scaled to have a gain of one at z=l. 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. The single row and 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 and 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, a trial and error method was used to first find a "diverging step size" which 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 4 significant digits.

Table 4.1 lists the results of the different simulations. Note that as the reference filter becomes more narrowband, the direct form takes much longer to adapt than either of the other two structures. As well, 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 high sampling frequency to 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=l. The graph in figure 4.9 summarizes the convergence times for the varying narrowband reference filters and different adaptive filter structures. These results indicate that using structures other than direct form can result in much better adaptation rates in oversampled applications.

4.7. Summary

An algorithm was presented for adapting a general state-space filter. This algorithm required N+l extra state-space filters to obtain all the gradients required for adapting an N'th

$\frac{\omega_s}{\omega_p}$	Transfer Function	Initial Poles		Direct Adapt	Row Adapt	Column Adapt
4	poles 0.5122 $0.06429 \pm j0.8625$ zeros -1.0 -0.675 lkj0.7377	0 0 0	Step Size μ	0.01	0.0025	0.0028
			Iterations for Convergence	16K	60K	50K
			Noise Measure N_M	0'8	8.2	8.2
8	poles 0.7714 $0.6920 \pm j0.5904$ zeros -1.0 $0.09597 \pm j0.9954$	0.7 0.7 0.7	Step Size μ	0.00028	0.03	0.015
			Iterations for Convergence	500K	60K	35K
			Noise Measure N_M	4.8	2.4	5.0
16	poles 0.8788 $0.8872 \pm j0.3379$ zeros -1.0 $0.6581 \pm j0.7529$	0.8 0.8 0.8	Step Size μ	0.000025	0.01	0.00375
			Iterations for Convergence	6MEG	50K	30K
			Noise Measure N_M	49.	1.6	2.8
32	poles 0.9375 $0.9574 \pm j0.1775$ zeros -1.0 $0.9019 \pm j0.4318$	0.9 0.9 0.9	Step Size μ	0.000001	0.0125	0.0015
			Iterations for Convergence	>10MEG	40K	40K
			Noise Measure N_M	652.	1.4	2.6

Table 4.1: Adaptation rates and noise measures for filters of varying bandwidths.

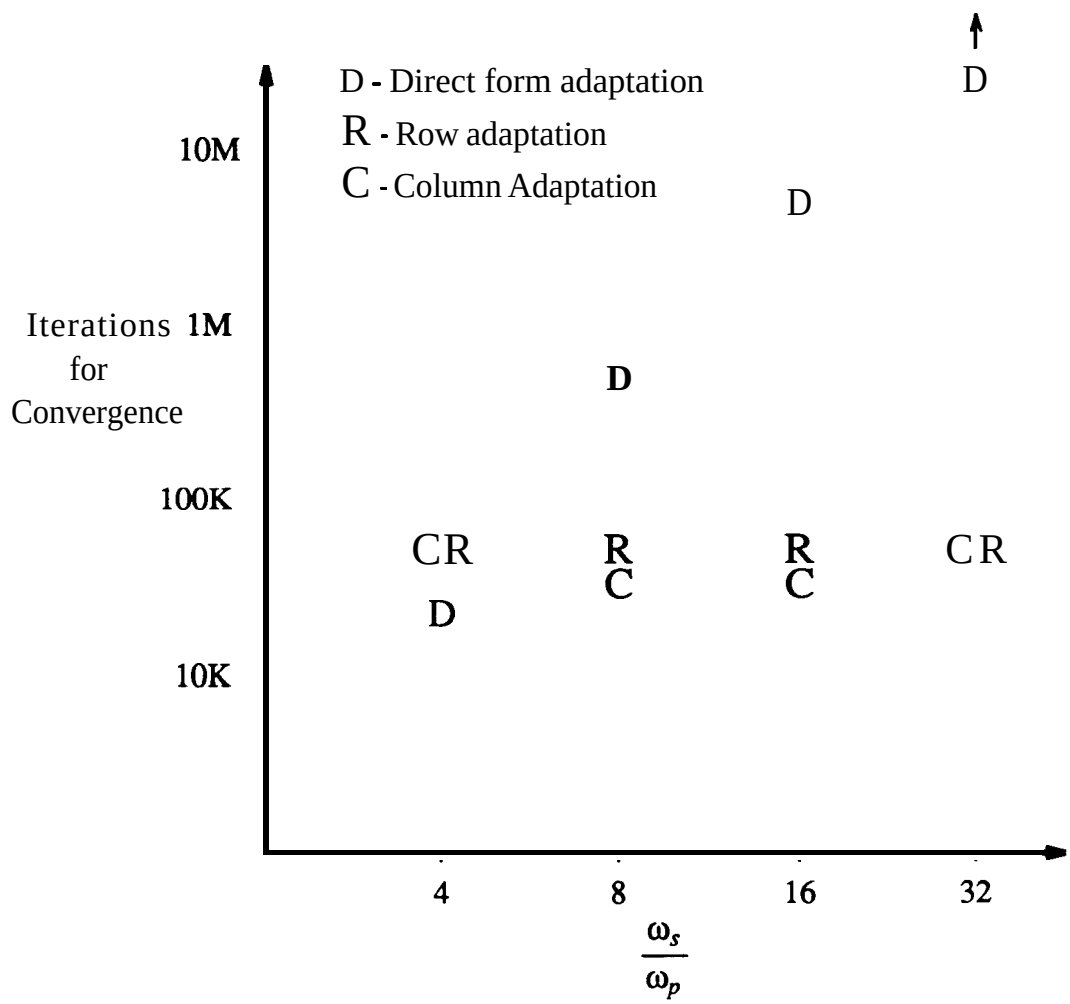


Figure 4.9: Convergence times of different structures for filters of varying bandwidths.

order system. Single column and single row adaptive filter structures were then introduced where only 1 extra state-space filter is 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 row adaptive filters can result in improved convergence rates and significantly better roundoff noise performance as compared to direct form implementations. These new adaptive filters are especially effective in the practical case of oversampled systems.

As a final comment, it should be pointed out 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 in this chapter can be easily modified to be applied in the continuous time domain. Adaptive filtering in the continuous time domain will be the main focus of the next chapter.

Appendix 4.A

Quasi-orthonormal design procedure

A state-space orthonormal design technique was presented for continuous-time circuits in chapter 2. 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. We present in this appendix 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

$$s\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}u \qquad \text{(4.A. 1)}$$
$$\mathbf{y} = \mathbf{c}^T\mathbf{x} + du$$

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

$$z\mathbf{x} = (\mathbf{A} + \mathbf{I})\mathbf{x} + \mathbf{b}u \qquad \text{(4.A.2)}$$
$$\mathbf{y} = \mathbf{c}^T\mathbf{x} + du$$

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:

(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 chapter 2 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 asymptotically become equal and the off-diagonal elements 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 while noise in continuous-time systems is defined as having unit power density over 1 rad/s. Thus a quasi-orthonormal system obtained as described above will asymptotically have mean-square output levels a factor 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 will always result in a sparse tridiagonal structure for the **A** matrix.

Appendix 4.B

Approximate algorithm interpretation

This appendix will relate the approximate algorithm presented in [Feintuch, 1976] to the algorithm proposed in this paper. The approximate algorithm does not require any extra computations to obtain gradient signals and is therefore used in practical applications [Eriksson and Allie, 1988].

Note that the algorithm in [Feintuch, 1976] was developed for a direct form filter where a zero-forming filter is cascaded with a pole-forming filter. The zero-forming filter is adapted using the usual LMS algorithm with true gradients so we will not concern ourselves with it. On the other hand, the pole-forming filter is adapted with gradients intended to approximate the true gradients.

The transfer function, $P(z)$, of the pole forming filter is

$$P(z) = \frac{z^{-1}}{1 - a_N z^{-1} - a_{N-1} z^{-2} - \cdots - a_1 z^{-N}} \tag{4.B. 1}$$

where a factor of z^{-1} has been introduced in the numerator to simplify the state-space representation. This transfer function is realized by the direct form state-space system

$$\mathbf{A} = \begin{vmatrix} 0 & & & 10.0 & & 0 \\ 0 & 0 & 1 & . & 0 & 0 \\ . & . & . & . & & . \\ 0 & 0 & 0 & . & i & 0 \\ 0 & 0 & 0 & . & 0 & 1 \\ a_1 & a_2 & a_3 & . & a_{N-1} & a_N \end{vmatrix} \qquad \mathbf{b} = \begin{vmatrix} 0 \\ 0 \\ . \\ 0 \\ 0 \\ 1 \end{vmatrix} \tag{4.B.2}$$

$$\mathbf{c}^T = [\,0\,0 \quad \cdots \quad 0\,1\,] \quad d = 0$$

It is not difficult to show that the adaptation algorithm of [Feintuch, 976] applied to the above state-space system results in the update equation

$$a_i(n+1) = a_i(n) + 2\mu e(n)x_i(n-1) \tag{4.B.3}$$

Comparing this result with that of equations (4.11) and (4.15), we see that the approximate gradient algorithm simply replaces the impulse response $g_N(n)$ by a single delay. It was shown in the simulation results that this approximation is good in some applications and poor in others. This interpretation helps to explain the adaptation paths taken by the algorithm presented in [Feintuch, 1976].

Chapter 5

Monolithic Implementation and Experimental Results

5.1. Introduction

One of the main contributions of this thesis is to demonstrate that monolithic analog adaptive IIR filters are feasible. To demonstrate that discrete analog adaptive IIR filters are realizable, a discrete third-order prototype was constructed and the design details are presented in section 5.2. The *single row* adaptive filter algorithm described in chapter 4 was chosen as the basis for the prototype, and although one of the motivations for developing analog adaptive filters is high frequency applications, for evaluation purposes, a low frequency prototype was constructed. The experimental results for this discrete prototype are given in section 5.3 and show that the algorithms presented in chapter 4 can successfully be converted to the analog domain. However, although the building blocks used in the discrete prototype are typical analog circuits that have been previously integrated, the specific implementation details of these building blocks are not intended for a monolithic realization. To show the feasibility of monolithic realizations, the design details and experimental results for a CMOS monolithic third-order programmable filter are described in sections 5.4 and 5.5. This programmable filter is realized using the transconductance-C technique with voltage signals that adjust the values of the filter coefficients by varying the transconductance of differential input circuits.

5.2. Discrete prototype design details

To realize the coefficient update algorithm in hardware, one requires some type of multiplier. Since high quality analog multipliers are difficult to implement, it was decided to avoid using standard multipliers in the coefficient update algorithm block by implementing the *sign-data* algorithm [Treichler et al, 1987] rather than the traditional LMS algorithm. With the sign-data algorithm, the error signal is multiplied by only the sign of the gradient signal rather than the gradient signal itself and therefore the multiplication can be realized by simple hardware. Specifically, the sign of the gradient signal can be determined by a comparator and with the use of a multiplexor, the multiplier output is set to either the inverted or non-inverted error signal depending on the comparator result. Since multiplexors do not introduce any significant offsets, this type of multiplication results in less output offset voltage than a simple multiplier circuit. Offsets at the coefficient update integrators can cause problems as will be shown in chapter 6.

The block diagram for the third-order single-row analog adaptive filter is shown in figure 5.1. The basic structure of the programmable and gradient filters is of the orthonormal ladder type described in chapter 3. There are six coefficients used to adjust the transfer function of the programmable filter; a_i ,i=l-3 and c_i ,i=l-3. As shown, three pole coefficient update blocks are used to adapt the a_i coefficients while the c_i coefficients are adapted using three zero coefficient update blocks. Note that these coefficient update blocks use the sign data algorithm discussed above. 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, $\alpha_i(t)$, are obtained from a gradient filter. Note that the gradient filter is identical to the feedback circuit used in tie programmable filter. The error signal is obtained as the difference between the programmable filter's output and an external reference signal, $\delta(t)$.

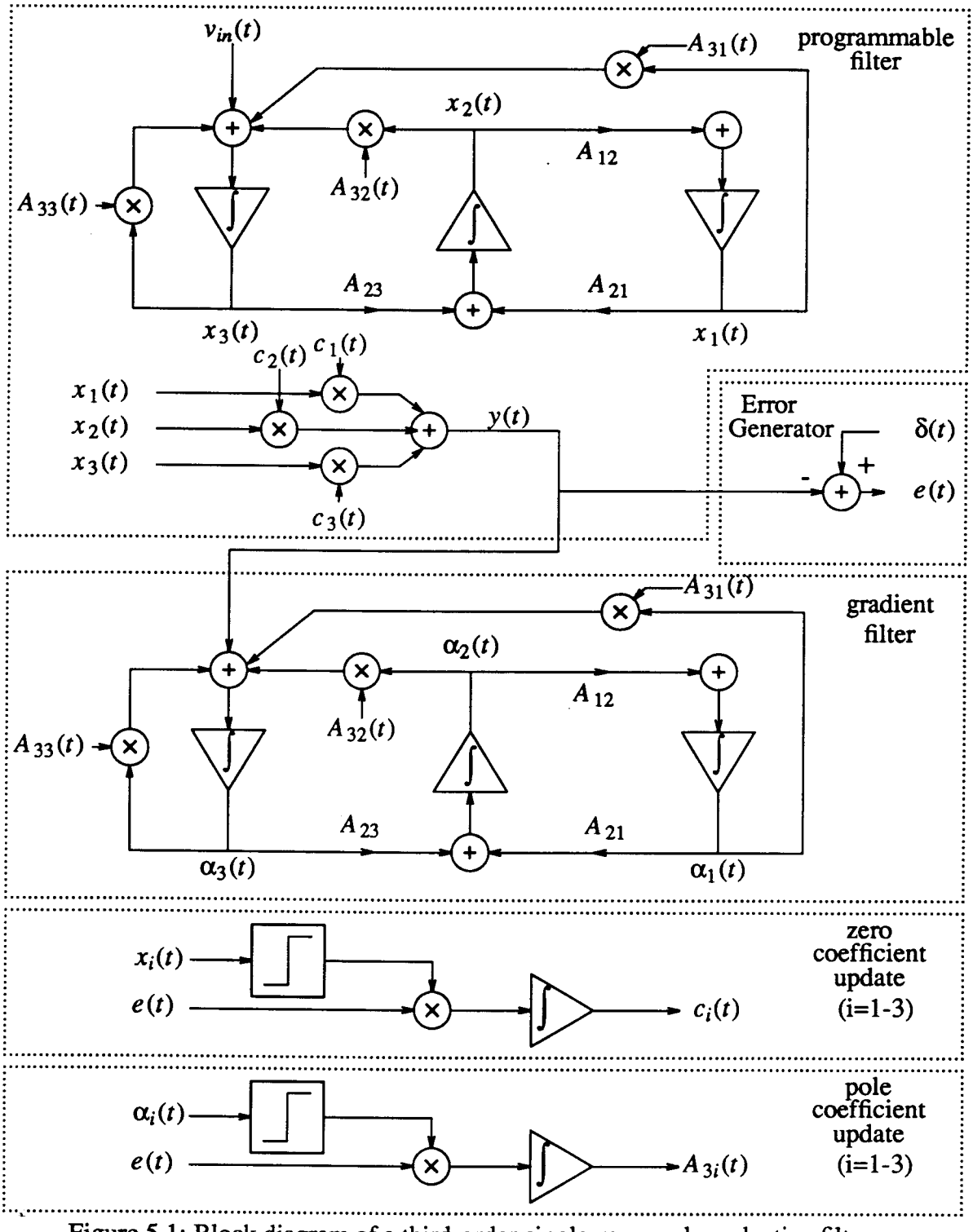


Figure 5.1: Block diagram of a third-order single-row analog adaptive filter

We see from the block diagram in figure 5.1 that multiplier/summer circuits are required for the programmable and gradient filters. The circuit that realizes these multiplier/summer stages is based on a circuit linearization technique that was originally proposed for creating fixed continuous-time integrated filters [Banu and Tsividis, 1983] where control over time-constants is necessary to account for process variations. The circuit realization for a single multiplier and summer stage is shown in figure 5.2. Matched N-channel MOS transistors are used to implement the four transistors and are contained in one Siliconix integrated circuit (the SD5001N).

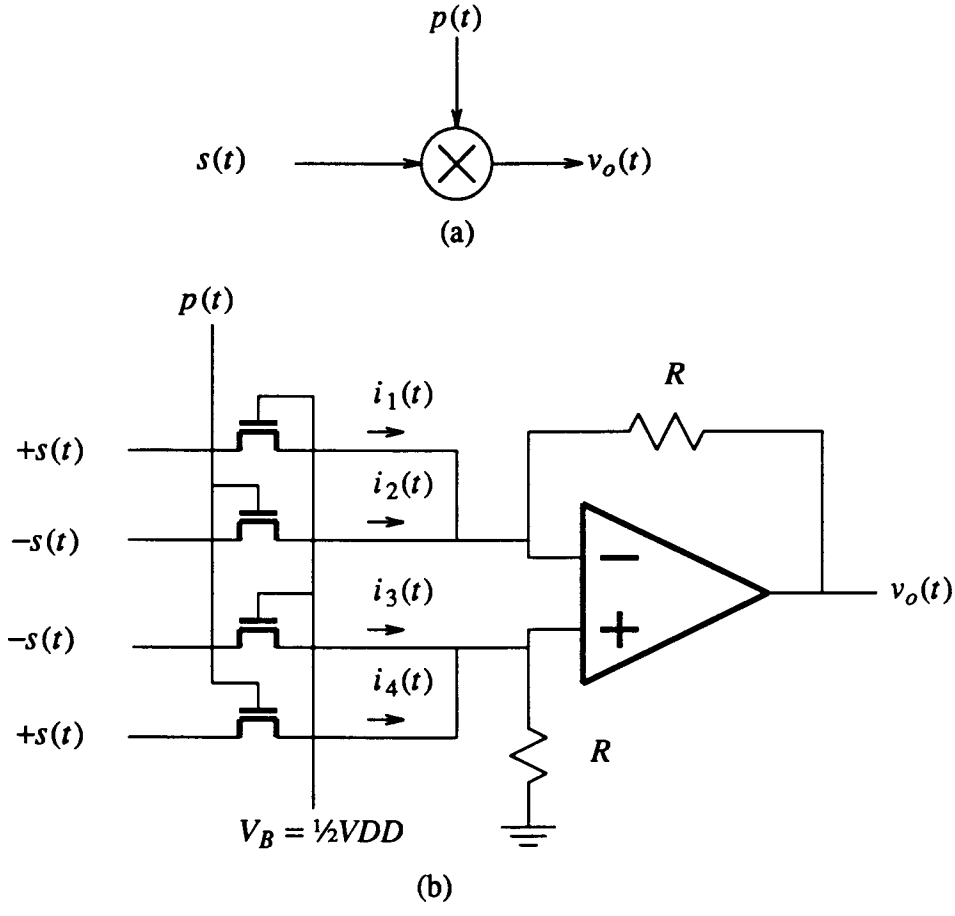


Figure 5.2: (a) Multiplier symbol (b) Circuit implementation

The voltage V_B is a DC bias voltage set to halfway between ground and the positive rail.

Referring to figure 5.2, the output voltage $v_o(t)$ is easily seen to be

$$v_o(t) = [\,(i_4(t) - i_2(t)) + (i_3(t) - i_1(t))\,]R \tag{5.1}$$

With reference to the term containing $i_4(t)$ and $i_2(t)$ *in* equation (5.1) above, one can show that through the use of balanced inputs, $+s(t)$ and $-s(t)$, and the fact that the op-amp input terminals are approximately equal in voltage, the resulting current difference consists of a linear term and odd-order nonlinear terms'. In other words, for the output voltage, the even-order nonlinear terms are cancelled (for a most complete description of this cancellation, the reader is referred to [Tsividis et al, 1986]). A similar distortion cancellation is obtained in the current difference term containing $i_3(t)$ and $i_1(t)$. Assuming the transistors operate in the triode region where the odd-order distortion products are small, the result is a linearizing effect such that the output signal has low distortion for signals as large as 1 volt peak.

To further analyze this multiplier circuit, we can write an equation for the output voltage as a function of the input signal, $s(t)$, and the coefficient signal, $p(t)$, using the linear term of each transistor. The linear resistance, r_{ds}, of a transistor in the triode region can be written as [Tsividis et al, 1986]

$$r_{ds} = \left[\mu_N C_{ox} \frac{W}{L}(V_G - V_T)\right]^{-1} \tag{5.2}$$

where μ_N is the N-channel mobility, C_{ox} is the gate oxide capacitance per unit area, W and L *are the* channel width and length, V_G is the gate potential and V_T is the threshold voltage. Using this formula, the output voltage, $v_o(t)$, can be written as

$$v_o(t) = 2R(\mu_N C_{ox} \frac{W}{L}) s(t)(p(t) - V_B) \tag{5.3}$$

[1] Note that a square-law model for the transistors would result in perfect linearity with this circuit.

From this formula, we see that this circuit acts as a multiplier of the two input voltages $s(t)$ and $p(t)$ to obtain the output voltage, $v_o(t)$. Note, there is an offset in the above formula such that p(f) must equal V_B for the output voltage to go to zero, however, this introduced offset does not affect the operation of the final realization. As well, note that both positive and negative gains can be obtained by having $p(t)$ above or below V_B, respectively, and that, ideally, the minimum gain obtainable is zero. However, there is a limit to the maximum gain of this same circuit determined by the resistor value R and the minimum r_{ds} obtainable.

Figure 5.3 shows the circuit implementation of the discrete prototype at a block level where only 4 different blocks are used to create the circuit with the details of each of these blocks shown in figures 5.4 to 5.7. This set of circuit drawings describe the complete realization for the discrete prototype analog adaptive filter. Comparing the block diagram in figure 5.1 with the circuit implementation blocks in figure 5.3, we see that the programmable filter consists of a "Variable Feedback" block and a "Variable Sum" block while the gradient filter consists of only a "Variable Feedback" block. As well, the error generator corresponds to the "Error Gen" block with each of the zero and pole coefficient update circuits consisting of a single "Coeff Update" block.

Let us now look at each of the blocks in more detail. The summer and three multipliers in the "Variable Feedback" and "Variable Sum" blocks (figures 5.4 and 5.5, respectively) are implemented using the N-channel MOS multiplier described above. In this implementation, transistor currents are summed to provide a single output signal corresponding to the necessary multiplication and summation of the input signals. The resistor values used in the multipliers/summer circuits were experimentally chosen to give maximum gains of about 5 for the gain from a single S_i signal to the output implying that coefficients values can range from -5

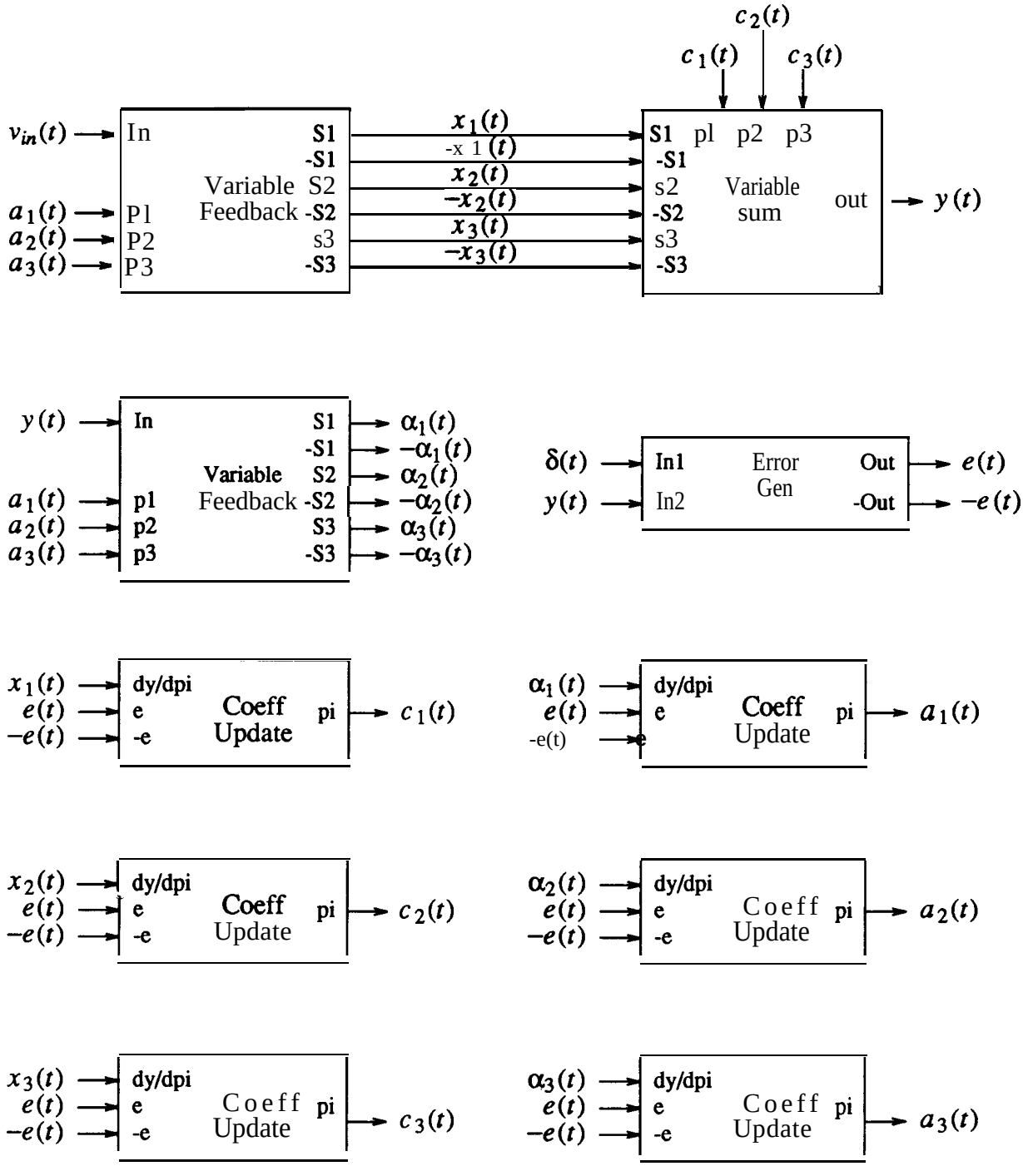
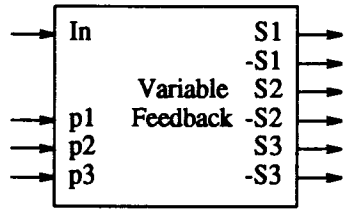
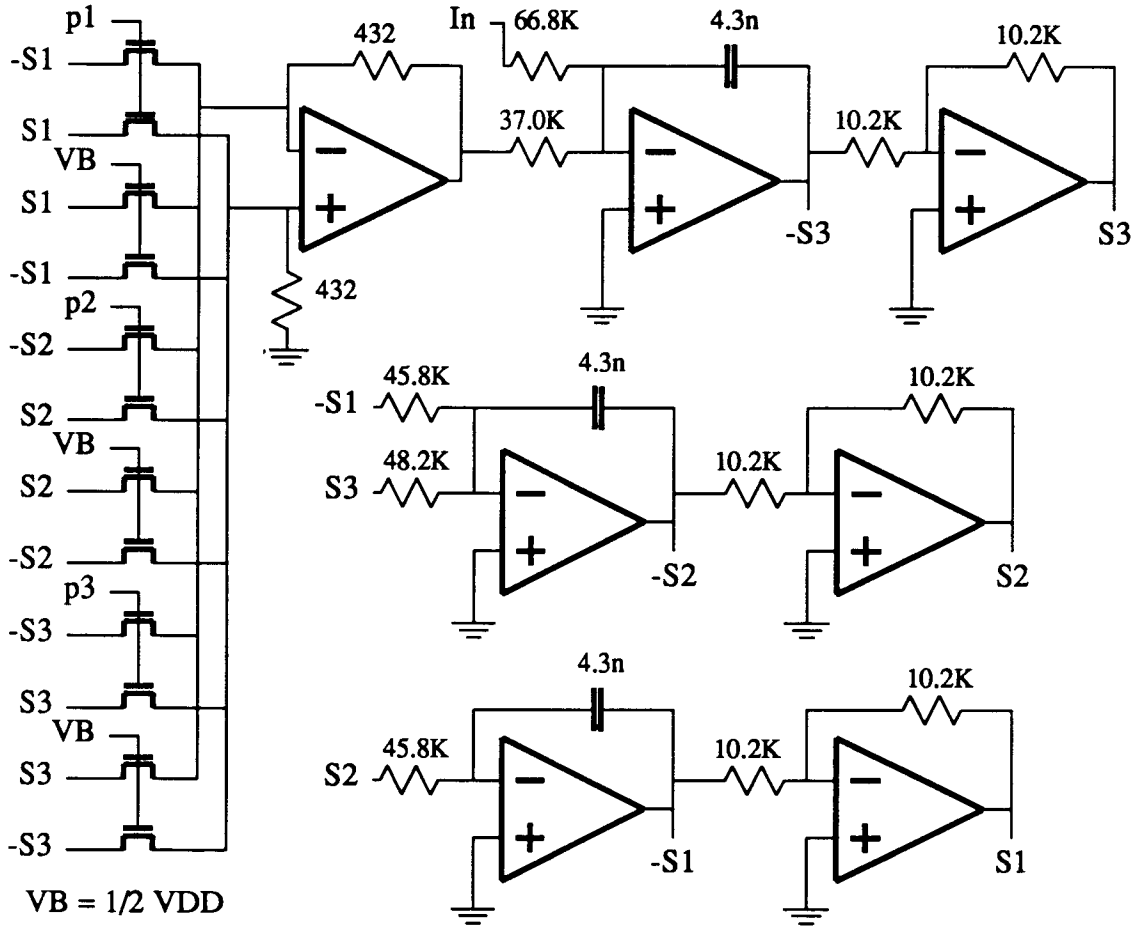


Figure 5.3: Circuit implementation of breadboard prototype at block level.

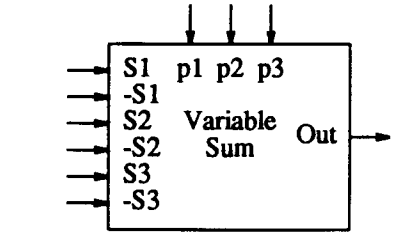


(a)

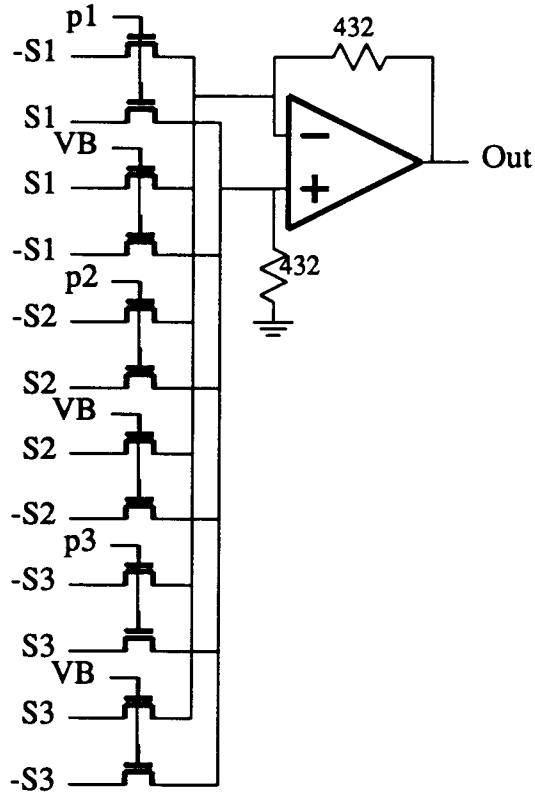


Notes: (1) All op-amps are TL082B
(2) N-channel transistors in SD5001 (b)

Figure 5.4: Circuit details of "Variable Feedback" block.
(a) Symbol (b) Circuit Implementation



(a)

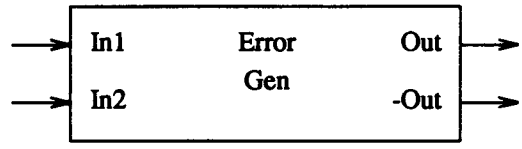


VB = 1/2 VDD

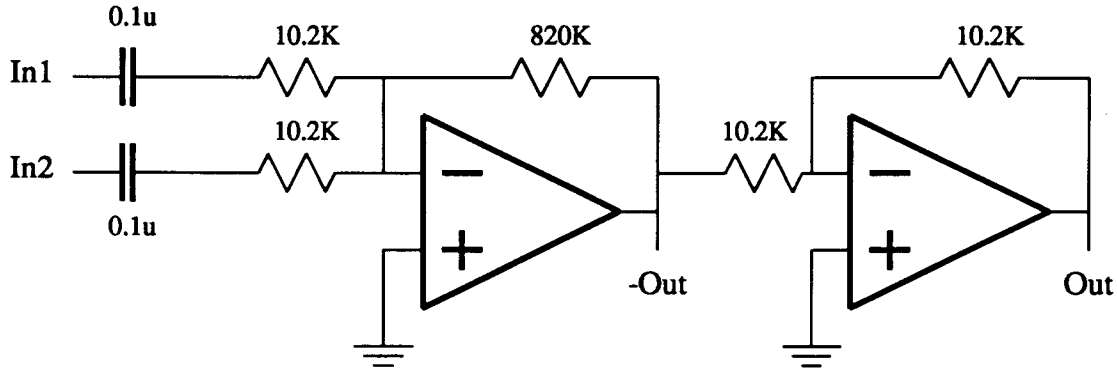
Notes: (1) Op-amp is 1/2 of a TL082B
(2) N-channel transistors in SD5001

(b)

Figure 5.5: Circuit details of "Variable Sum" block.
(a) Symbol (b) Circuit implementation



(a)



Notes: (1) Op-amps are TL082B

(b)

Figure 5.6: Circuit details for "Error Gen" block.
(a) Symbol (b) Circuit implementation

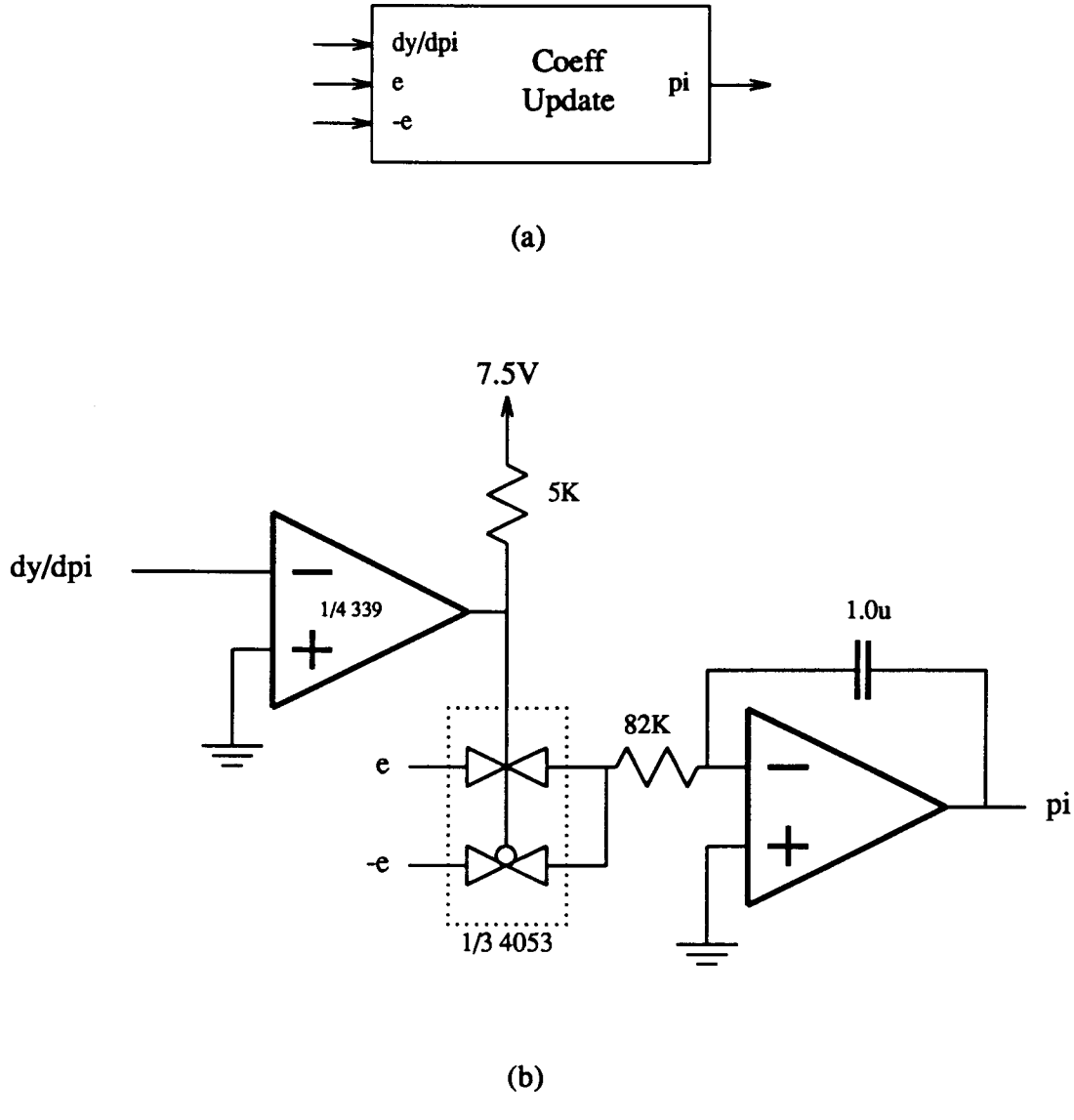


Figure 5.7: Circuit details of "Coeff Update" block.
(a) Symbol (b) Circuit implementation

to 5. The remaining circuitry in the "Variable Feedback" block corresponds to an implementation of the feedback network for the programmable or gradient filters. Looking at the "Error Gen" block in figure 5.6, we see that it directly implements the error generation block diagram in

figure 5.1 except that the inputs are summed rather than subtracted and the inputs are AC coupled and amplified. A summation rather than subtraction of inputs is used since it requires less circuitry and although using a summation inverts the sign of the gradient signal, this effect can easily be accounted for in the coefficient update circuitry. The reason for the AC coupling and amplification of the error signal is to reduce the effect of DC offsets (DC offset effects are discussed in chapter 6). It should be pointed out that on an IC realization of an analog adaptive filter, this AC coupling could be accomplished by using an extra summing coefficient to cancel out the DC offset in the error signal. Finally, referring to the circuitry for the "Coeff Update" block in figure 5.7, note that the multiply in the "Coeff Update" block is performed using a two input multiplexor since the the sign-data algorithm was implemented, as discussed above.

Since single-row adaptive filters normally require some estimate of final pole locations, it was decided to choose component values for the programmable and gradient filters so that the fixed time constants corresponded to values used in the reference filter that was used in the first experimental results discussed below. Thus, the following state-space filter was implemented.

$$\mathbf{A} = \begin{bmatrix} 0 & 0.98361 & 0 \\ -0.98361 & 0 & 1.2307 \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0.7737 \end{bmatrix} \tag{5.4}$$

$$\mathbf{c}^T = [\, c_1 \quad c_2 \quad c_3 \,] \quad d = 0$$

The coefficients in the above matrices which are shown as variables correspond to the state-space coefficients which are adapted. This normalized state-space system was denormalized to the resistor and capacitor component values shown by multiplying all coefficient values by $2\pi \times 10^3$. This gives time constant values around the 1 KHz range.

5.3. Discrete prototype experimental results

In order to test the adaptive filter, a model matching application was used where the reference signal, $\delta(t)$, is obtained as the output of a fixed filter with white noise applied to its input. The same white noise source is also applied to the adaptive filter input causing the adaptive filter to match the transfer function of the reference filter.

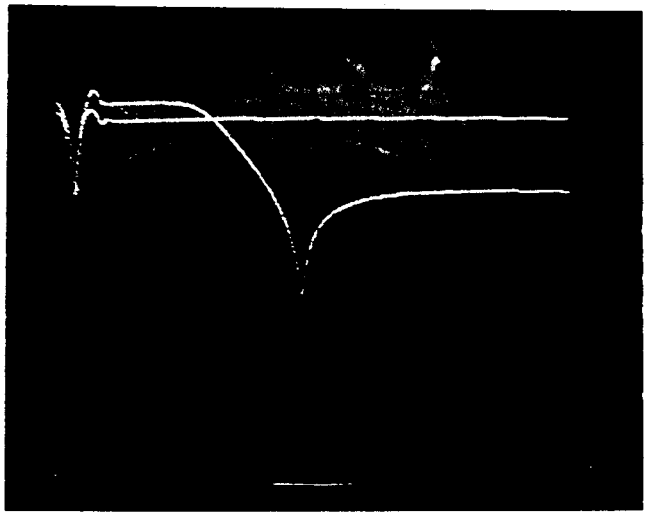
For the first experimental example, the reference filter was chosen to be a third-order lowpass filter with finite transmission zeros. The normalized state-space system for the reference filter was

$$\mathbf{A} = \begin{bmatrix} 0 & 0.98361 & 0 \\ -0.98361 & 0 & 1.2307 \\ 0 & -1.2307 & -1.8805 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0.7737 \end{bmatrix} \tag{5.5}$$

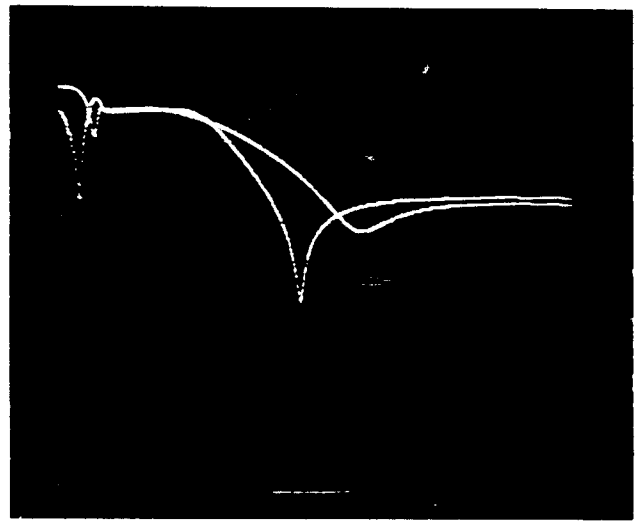
$$\mathbf{c}^T = [\ 1.5779\ 0\ 0.4563\ ] \quad d = 0$$

This normalized system was denormalized using the same scaling value $(2\pi \times 10^3)$ as the programmable and gradient filters and resulted in the passband edge at 1 KHz. Note that, except for the coefficients which adapt, this system is the same as that for the programmable and gradient filters and therefore, after adaptation, the coefficients A_{31}, A_{32}, and A_{33} should correspond to 0, -1.2307, and -1.8805, respectively and c_1, c_2, and c_3 should correspond to 1.5779, 0, and 0.4563, respectively. Therefore, 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 that will be presented.) For this example, the normalized reference filter has a pole at the location -1.1167 and a pair of complex poles at $-0.3819 \pm j\ 1.2179$. For the complex pair of poles, defining ω_0 and pole Q to be the natural frequency and pole Q, respectively, we find $\omega_0 = 1.2764$ and $Q = 3.34$.

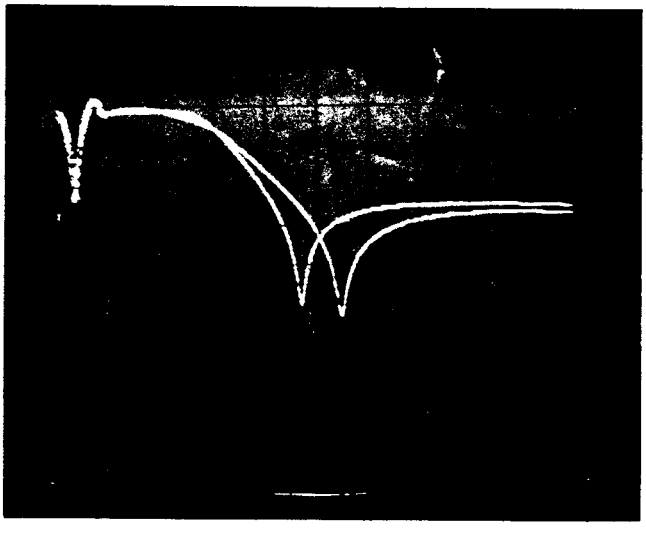
Figure 5.8 shows the adaptation process of the discrete prototype when using the reference filter described in equation (5.5) above. In order to observe the adaptation process, adaptation



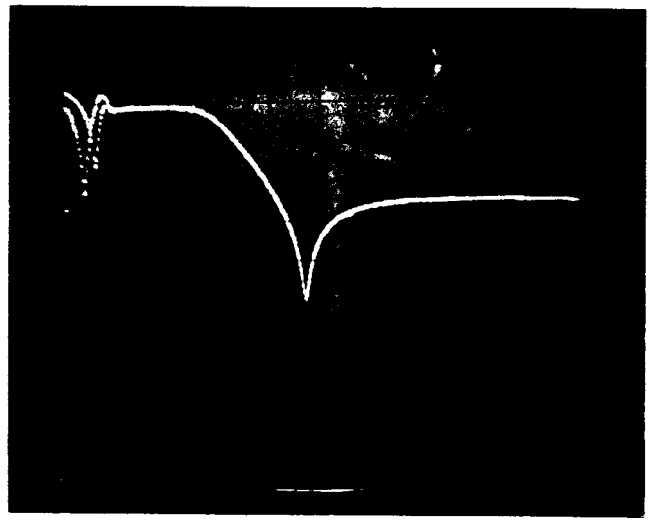
(a)



(b)



(c)



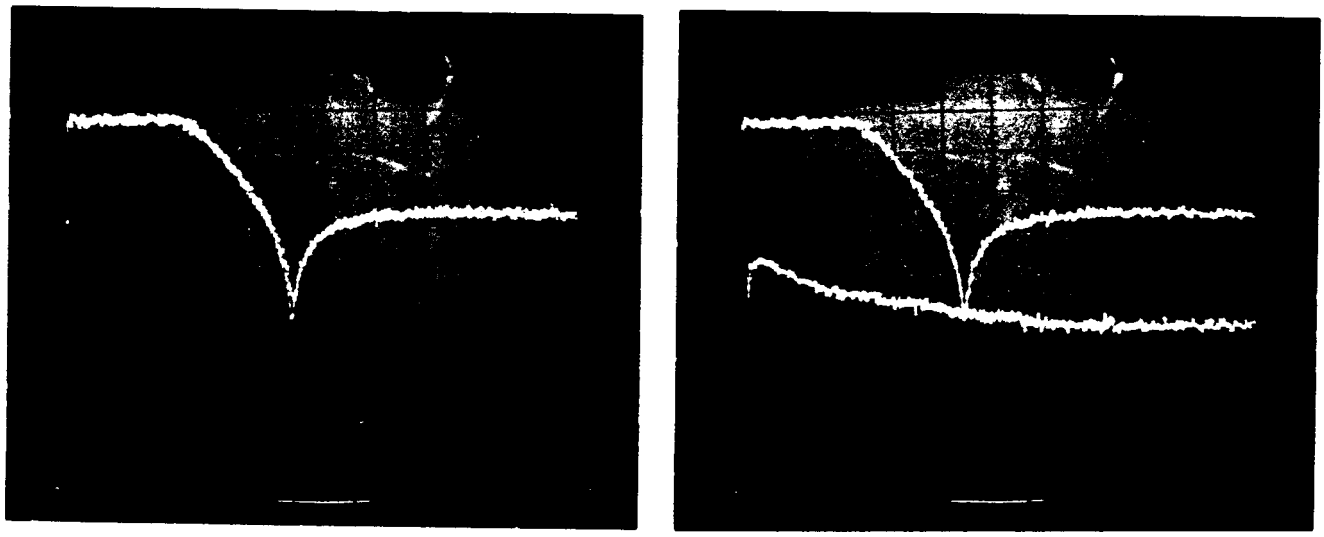
(d)

Figure 5.8: Experimental results for adaptive filter model matching application.
Vertical scale = 10 dB/div Horizontal scale = 500 Hz/div
(a) At power up. Horizontal line is adaptive filter's transfer function.
(b) At 7 sec (c) At 6 sec

speed was deliberately set quite slow and the swept sinusoid output from the spectrum analyzer was used to approximate a white noise input. Note that the unusual looking part of the transfer-function curve seen near low frequencies is a result of using the fast swept sinusoid as the system input and is seen on both the reference filter's response as well as the adaptive filter's response. Figure 5.8(a) shows the transfer function of the reference filter and the initial adaptive filter's transfer function at power up whereas figure 5.8(d) shows the same two transfer functions after adaptation is complete. Note that the two curves in figure 5.8(d) are almost identical as desired.

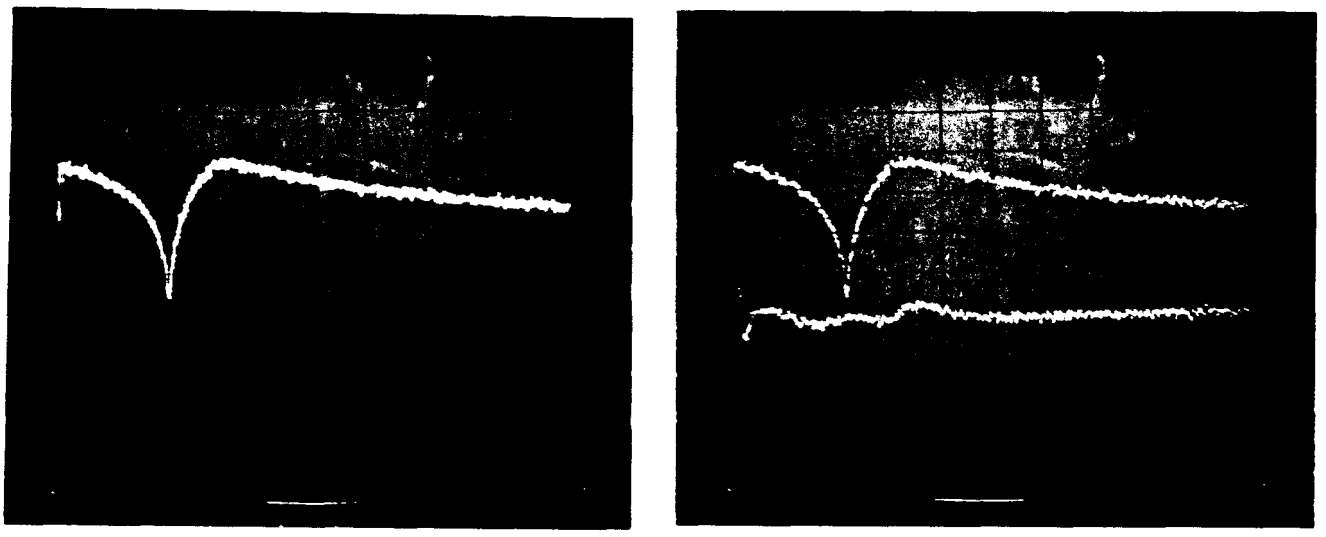
Referring to the adaptation times given in figure 5.8, the adaptation speed is seen to be quite slow. This adaptation speed can be increased so that adaptation occurs in under 1 sec by using a white noise generator as the signal input. Using a white noise generator for the signal input, figure 5.9(a) shows the spectrum of the outputs of the reference and adaptive filters where, as before, the two curves are difficult to distinguish. To determine the level of mismatch between the two spectra, the spectrum of the error signal is plotted along with the spectrum of the reference signal in figure 5.9. Note that the error signal is approximately 40 dB below the level of the reference signal indicating a close level of matching between the reference signal, $\delta(t)$, and the adaptive filter output, $y(t)$.

In figure 5.10, a different adaptation example is shown where the poles of the third order reference filter were changed to give a notch type transfer function. The components in the adaptive filter were unchanged and therefore remained optimized for the pole locations of the previous example implying that this example corresponds to the more realistic case where the pole locations can only be estimated and not known exactly. For this example, the reference filter had the state-space coefficients equal to



(a) (b)

Figure 5.9: Experimental results for third order lowpass 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)

Figure 5.10: Experimental results for third order notch 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 $e\,(t)$

$$\mathbf{A} = \begin{bmatrix} 0 & 0.98361 & 0 \\ -1.2307 & 0 & 0.98361 \\ 0 & -1.8805 & -1.2307 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0.7737 \end{bmatrix} \tag{5.6}$$

$$\mathbf{c}^T = [\,0 \;\; 0 \;\; 0.45631 \quad d = 0$$

The poles for this system are at the locations -0.5548 and $-0.3379 \pm j$ 1.6034 and have $\omega_0 = 1.6387$ and $Q = 4.85$ for the pair of complex poles. Note that these poles are significantly different than the previous example yet the adaptive filter successfully matched the reference filter.

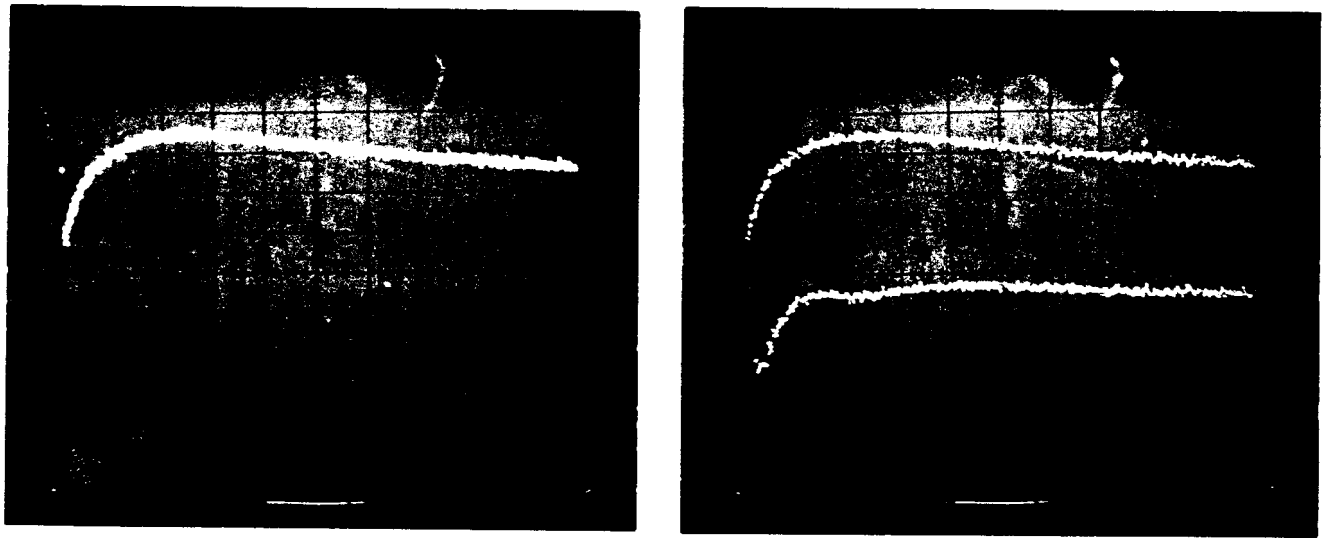
In a final example, the third order reference filter had the state-space coefficients equal to

$$A=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1.2307 \\ 0 & -1.2307 & -1.8805 \end{bmatrix} \quad \mathbf{b}=\begin{bmatrix} 0 \\ 0 \\ 0.7737 \end{bmatrix} \tag{5.7}$$

$$\mathbf{c}^T=[\,0 \;\; 0 \;\; 1.5779\,] \quad d=0$$

This system is only second order thus requires the third order adaptive filter to match a lower order system. The poles of this system are at the locations $-0.94025 \pm j\,0.79407$ with a zeros at 0 and ∞. The adaptation results for this example are shown in figure 5.11.

For this example, one might ask the question as to how the third order system successfully models a second order system. To answer this question, a computer simulation was run using the



(a) (b)

Figure 5.11: Experimental results for second order 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 $e(t)$

same parameters as this example. The final adapted system had the state coefficients

$$\mathbf{A} = \begin{bmatrix} 0 & 0.98361 & 0 \\ -0.9836\,1 & 0 & 1.2307 \\ 1.0337 & -2.0307 & -2.9185 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0.7737 \end{bmatrix} \tag{5.8}$$

$$\mathbf{c}^T = [\ -1.2611 \quad 1.3309 \quad 1.5779\ ] \quad d\ 0$$

The poles for this system are located at $-0.94025 \pm j\,0.79407$ and -1.0380 with zeros located at 0, ∞, and -1.0380. Note that a zero and pole are located on top of one another causing the 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 in the simulation or in the discrete prototype experimental results. In fact, the discrete prototype was left running for well over an hour and no instability was observed.

In concluding this section, these three examples show that the adaptive algorithms presented in chapter 4 can successfully be translated into analog designs. Note, however, that only a model matching application with a white noise input has been demonstrated in the three examples and that no uncorrelated noise was added during the experimentation.

5.4. Monolithic implementation design details

In this section, the design details for a monolithic implementation of a voltage controlled programmable continuous-time filter will be presented. The circuit was fabricated using a 3 micron CMOS process that is available to Canadian universities through the Canadian Microelectronics Corporation (CMC). Although the implemented circuit has both the programmable and gradient filters on chip, only the programmability aspects of the realization will be discussed in this thesis. Work is presently being done to build external circuits so that the integrated circuit can perform as an adaptive filter.

To construct a continuous-time programmable filter, one must first choose a basic technique to implement the filter. Two techniques dominate CMOS continuous-time filter implementations; MOSFET-C [Tsividis et al, 1986] and G_m–C (also referred to as transconductance-C) [Khorramabadi and Gray, 1984]. Of these two techniques, the MOSFET-C approach does not perform as well at high frequencies but appears to have a better dynamic range. Although dynamic range is important, it was felt that high frequency performance is a more important criterion for analog adaptive filters since this region of frequencies is where analog circuits have a distinct advantage over digital realizations. One of the reasons MOSFET-C implementations do not perform as well at high frequencies is the difficulty in designing high frequency op-amps that have the ability to drive resistive loads. However, with the G_m–C approach, only capacitive loads are driven and therefore one can make use of the design techniques developed for use in high-frequency switched-capacitor filters. In particular, a single-stage folded-cascode transconductance amplifier having an excellent high-frequency response can be used.

The circuit for a folded-cascode transconductance amplifier is shown in figure 5.12. The input differential pair of transistors connected to V_1 and V_2 converts the differential input signal, V_1–V_2, to a pair of currents, I_1 and I_2 which are then reflected in the output stage of 8 transistors such that the output node current, I_{out} is equal to I_1–I_2. The purpose of using stacked current mirrors in the output stage is to create a very high output impedance and therefore approach an ideal transconductance amplifier.

One drawback to this simple circuit is that the output current is not linearly related to the input differential voltage because of the nonlinear behavior of the input differential transistor pair. This disadvantage could be overcome by using one of the many interesting circuit techniques for realizing linearized transconductance circuits [Seevinck and Wassenaar, 1987]

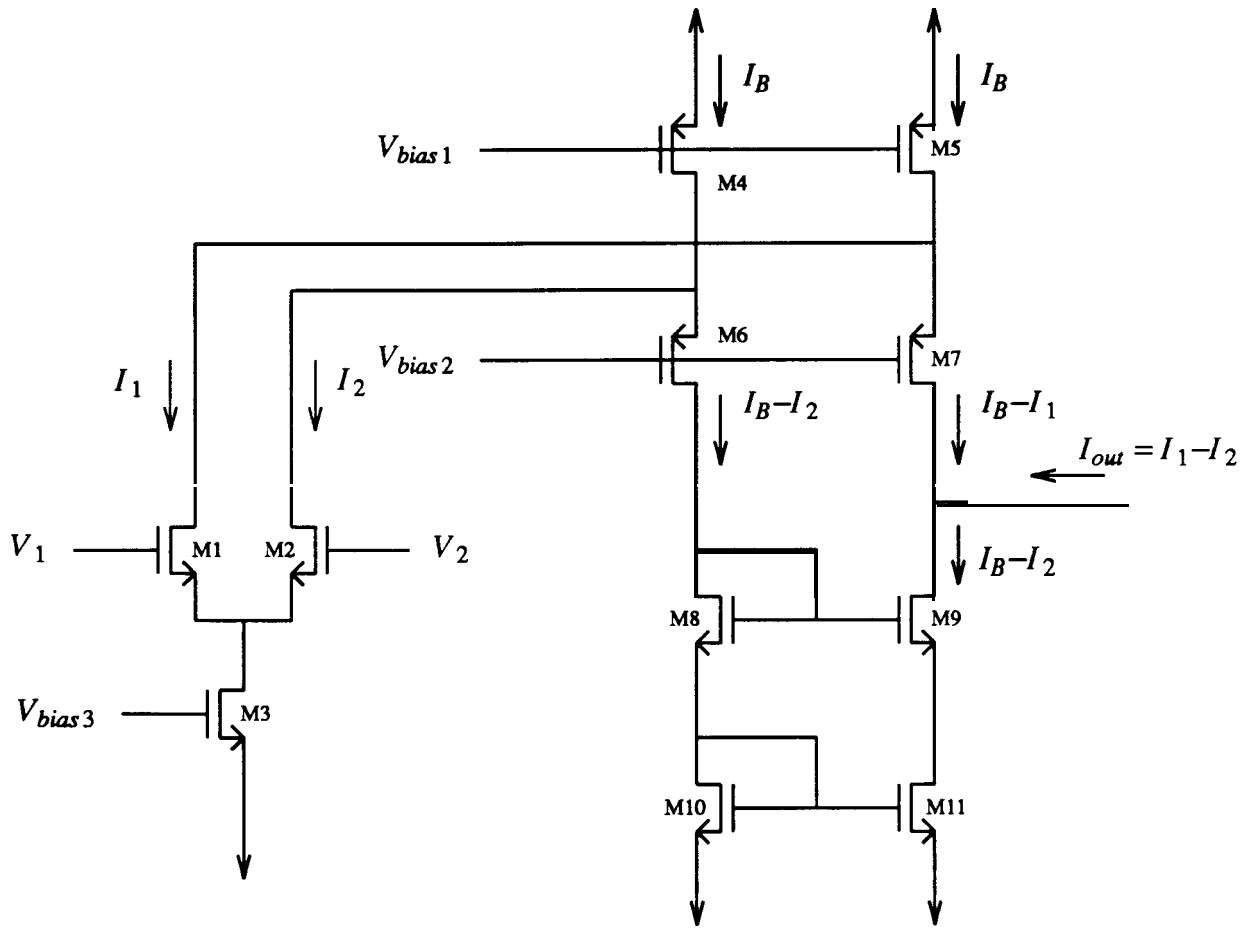


Figure 5.12: A folded-cascade CMOS transconductance amplifier

[Viswanathan, 1986] [Negungadi and Viswanathan, 1984]. However, since all these approaches increase the number of transistors required in the input stage, (and as we shall see, a large number of input stages is required) it was decided to build the prototype programmable filter using nonlinearized differential transistor pairs for the input stages.

The first step in the design was to develop a variable transconductance differential input stage where a single input voltage could continuously vary the transconductance of the stage from positive to negative values. One design for such an input stage is shown in figure 5.13.

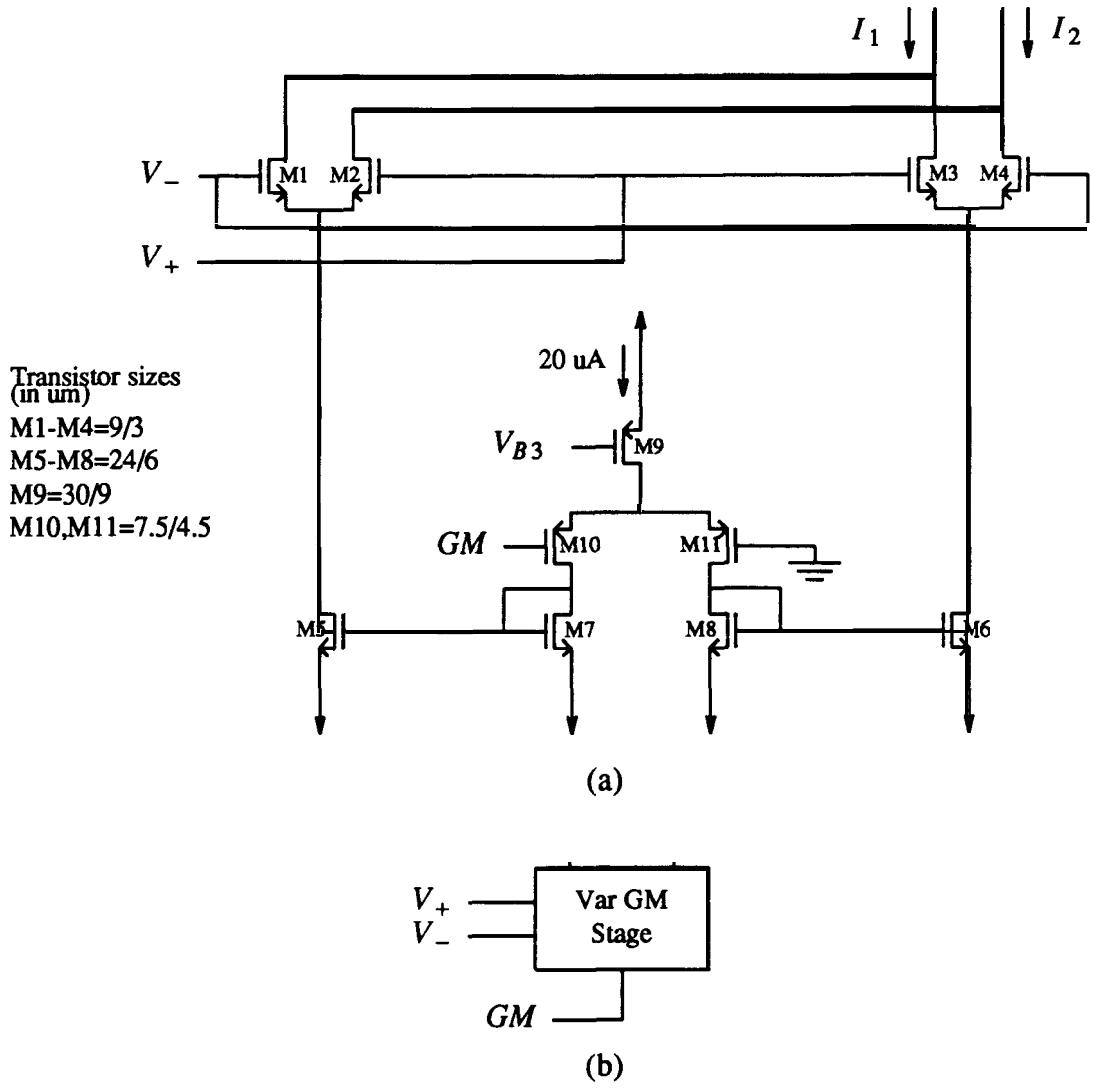


Figure 5.13: Positive and negative variable transconductance differential input stage. (a) Circuit details (b) Symbol used to depict circuit.

Defining $I_1 - I_2$ to be the output current, and $V_+ - V_-$ to be the input voltage, a positive voltage on *GM* results in a positive transconductance while a negative *GM* voltage causes a negative transconductance. When the transconductance control signal, *GM,* is at ground potential, both differential pairs of transistors receive the same bias currents and as a result of the cross coupling of these stages, the transconductance is zero. Neglecting the early voltage effect in the transistors, an interesting property of the circuit is that the sum of the output currents I_1 and I_2 is a constant value as the transconductance is varied. The transistors sizes shown in figure 5.13 are those used in the actual implementation of the programmable filter.

The basic circuit building block used to implement the programmable filter is shown in figure 5.14. This building block is essentially a transconductance folded cascode amplifier having three variable transconductance stages. The currents in the three variable GM stages are summed together to create a single pair of differential currents which is converted to a single output current by the transistors Ml-M8 in figure 5.14. The number of variable transconductance stages used in the building block is three since, as will be seen, this is the maximum number of signals summed into any one integrator in creating the programmable filter. If an integrator sums less than three signals, then the inputs into one or more of the variable GM stages are grounded. As well, note that the constant quiescent current into each of the variable GM stages make for a simple design in the folded cascode output stage transistors, Ml-M8, in figure 5.14. As before, the transistor sizes shown are those used in the implementation of the programmable filter. For completeness, the bias circuit that provides the three bias voltages to the transconductance amplifiers is shown in figure 5.15.

For the structure of the programmable filter, it was decided to use the transposed orthonormal structure described in chapter 3. A complete diagram of the programmable and gradient

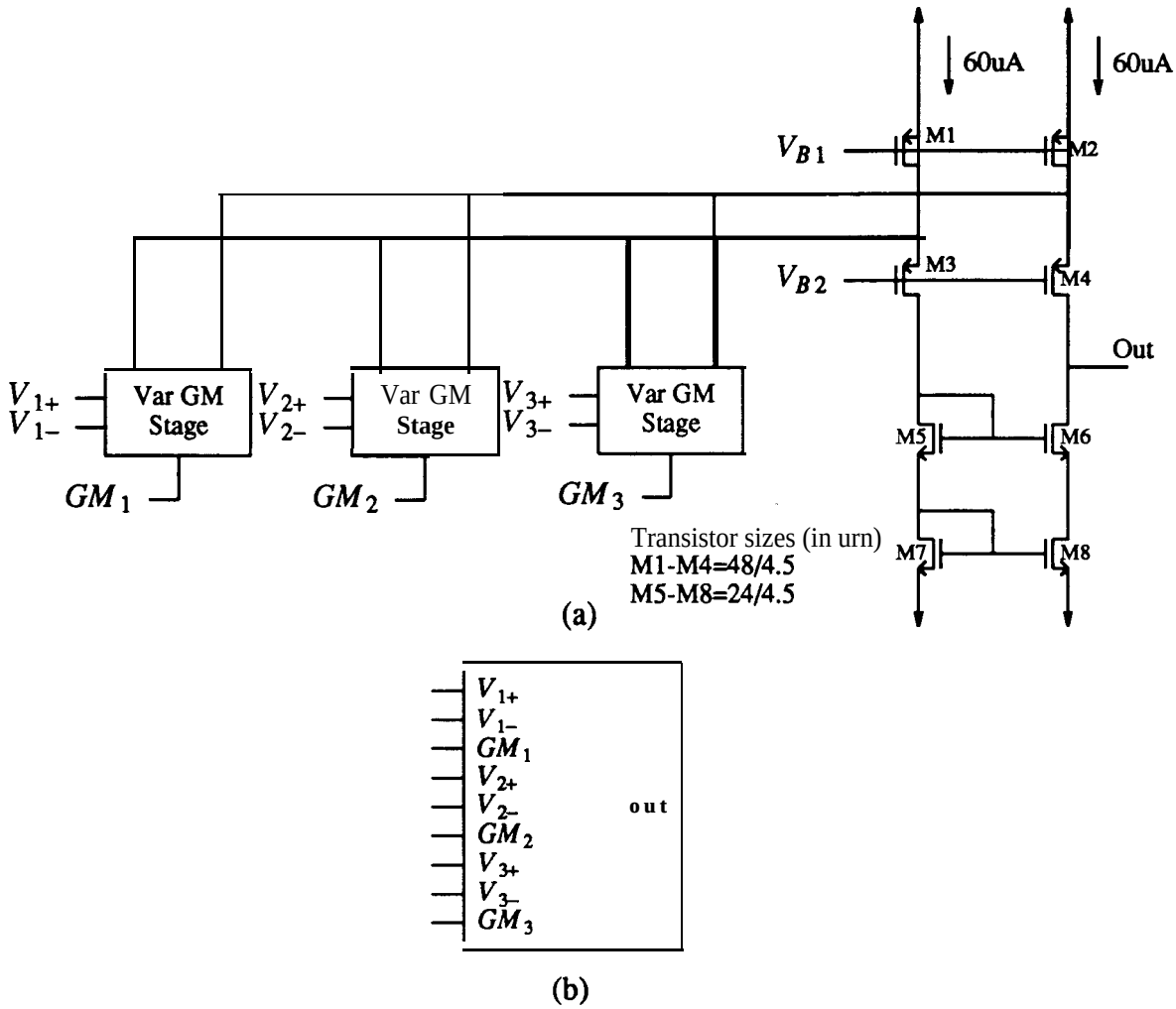


Figure 5.14: Three input variable transconductance folded cascode amplifier (a) Circuit details (b) Symbol used to depict circuit.

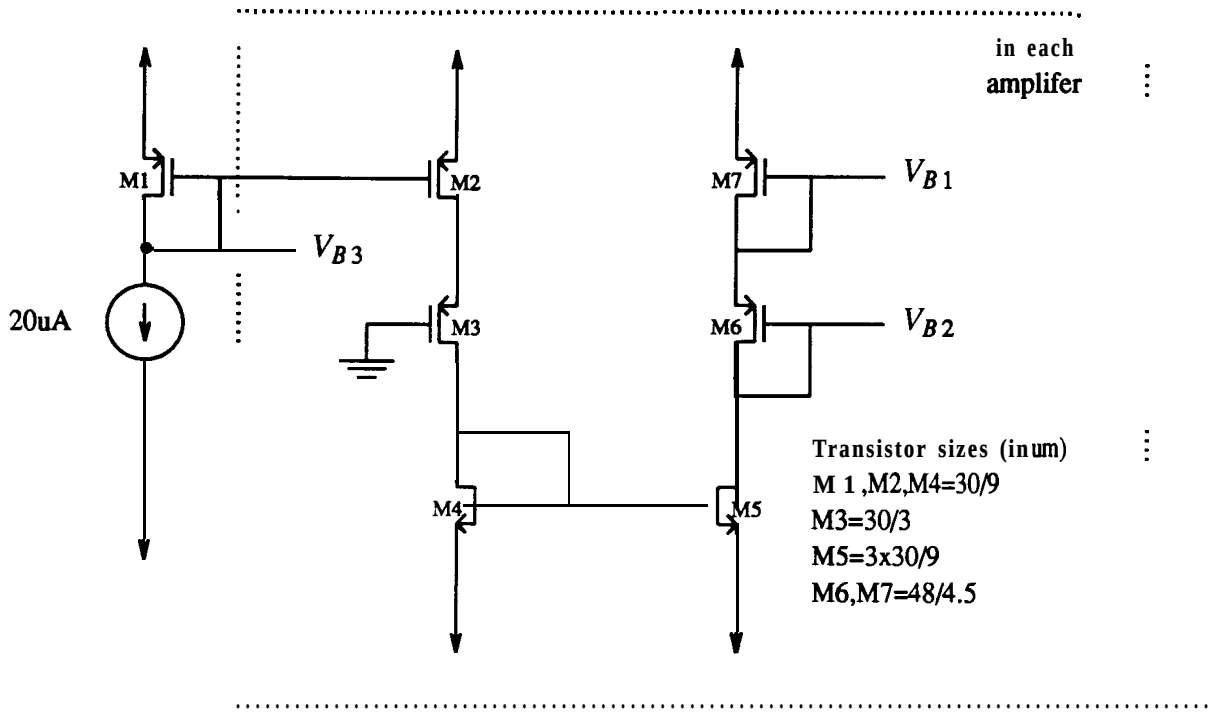


Figure 5.15: Bias circuitry for variable transconductance amplifiers.

filters using the basic three-input transconductance amplifier is shown in figure 5.16. With this circuit implementation, the following state-space system for the programmable filter is realized.

$$\mathbf{A} = \begin{bmatrix} 0 & -\alpha_1 & 0 \\ \alpha_1 & 0 & -\alpha_2 \\ 0 & \alpha_2 & -\alpha_3 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{5.9}$$

$$\mathbf{c}^T = \begin{bmatrix} 0\,0\,1 \end{bmatrix} \quad d = 0$$

The gradient filter realizes the transposed system of the programmable filter in order to generate gradient signals as described in chapter 4. Referring to figure 5.16 and the system in equation (5.9), the voltage inputs ALF1, ALF2, and ALF3 are the GM control inputs that adjust the system coefficients, α_1, α_2, and α_3, respectively. Similarly, the voltage inputs B 1, B2, and B3 adjust the coefficients b_1, b_2 and b_3. The output signals Xl, X2, and X3 correspond to the

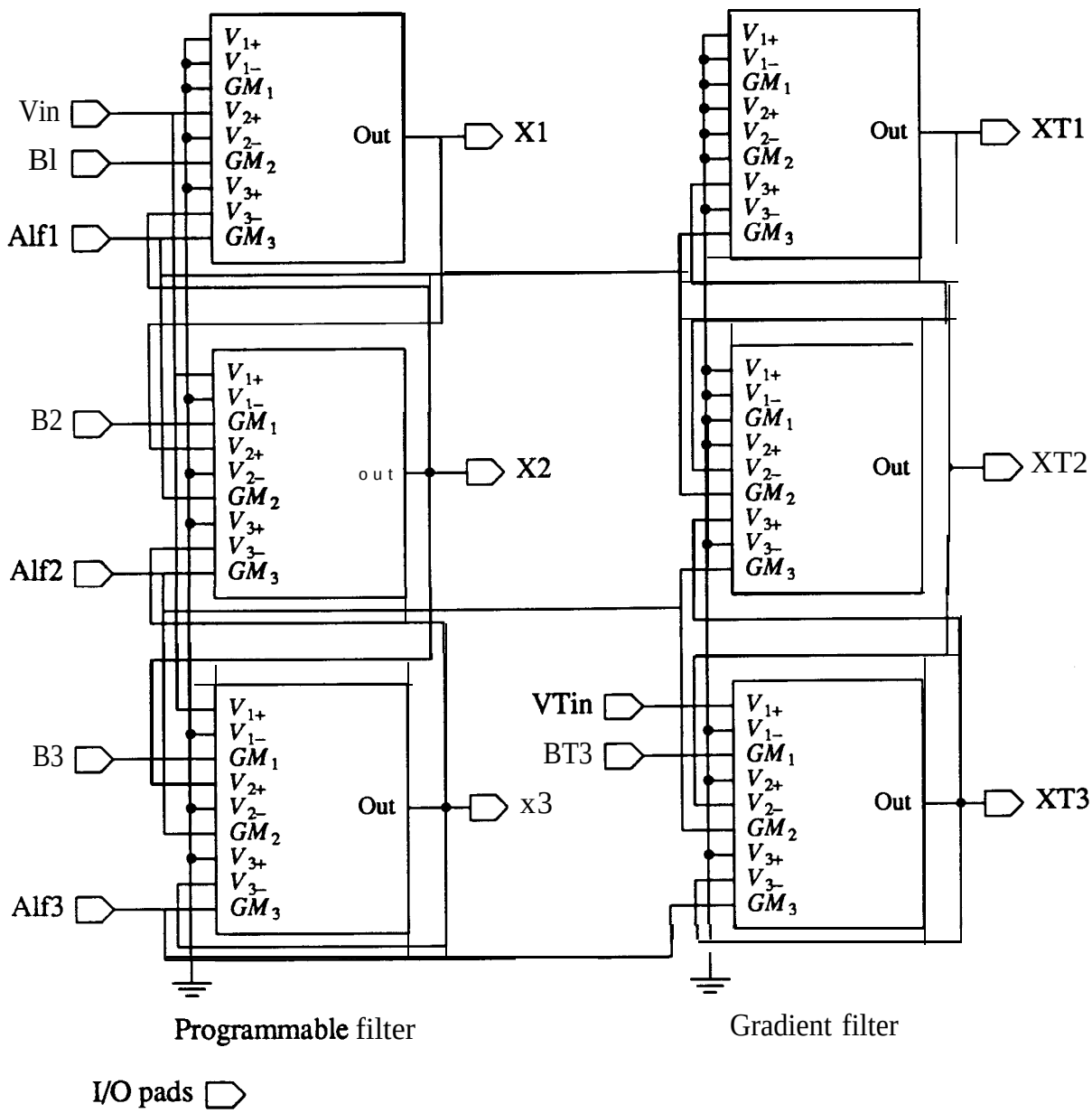


Figure 5.16: Orthonormal programmable filter implemented on prototype chip.

states (or, equivalently, the integrator outputs) of the programmable system whereas the signals XT1, XT2, XT3 correspond to the states of the gradient system. Note that from the state-space

system in equation 5.9, X3 is also the filter's output.

Referring to the system in equation (5.9), note that no output summing of states is used to create the zeros of the transfer function. Instead, input summing creates the necessary zeros through varying the coefficients b_1, b_2, and b_3 where these b vector coefficients are adjusted using the pad inputs B_1, B_2, and B_3 in figure 5.16. The reason that input summing (which is a result of using the *transposed* orthonormal structure) is used rather than output summing is that with output summing, a summing network would be required having a larger bandwidth than the integrator circuits and, unfortunately, this type of network is difficult to design. It should be pointed out here that if the programmable filter were of an order higher than three, say N, the use of input summing would still require at most three signals summed into any one integrator whereas output summing would require a separate summing stage having N signals summed together. Note also, that no capacitors are on chip as it was decided to use off-chip capacitors so that a slower prototype circuit could be evaluated.

5.5. Monolithic programmable filter experimental results

A photo-micrograph of the fabricated programmable and gradient filters is shown in figure 5.17 where the top right circuit is a test structure of the three input variable transconductance amplifier and the two lower circuits consist of the programmable and gradient filters (the upper circuit is the programmable filter). The area for the programmable filter is only 0.7 mm^2 and a similar area is taken up by the gradient filter. To bias the circuit, a bias resistor is placed between the bias pin and the negative supply. For the experimental results in this section, this bias resistor was chosen to be 300 KΩ and thus set the bias currents to approximately those shown in the circuit diagrams of the prototype. With a ± 3 volt power supply, the programmable filter circuit dissipates 3 mW or, equivalently, 1 mW per pole while the gradient filter dissipates

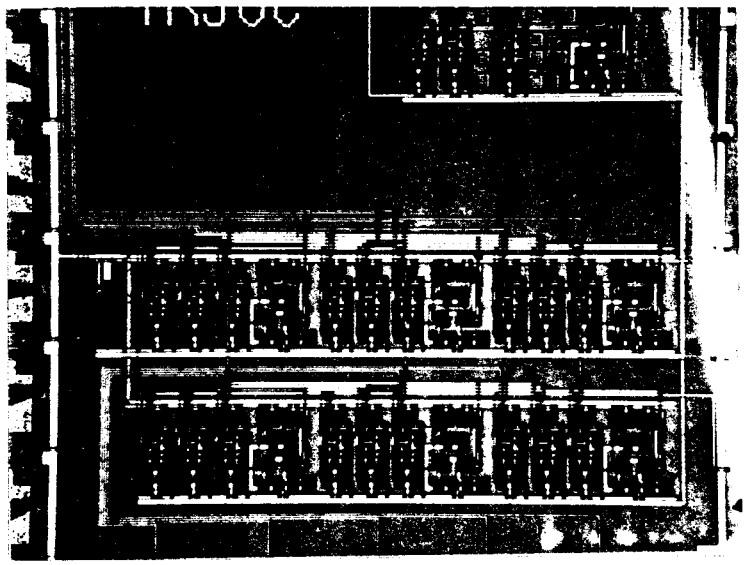


Figure 5.17: Photo-micrograph of 3 um CMOS programmable and gradient filters.

an equal amount of power.

The test structure of the basic amplifier was used to plot the transconductance of the amplifier vs. the control voltage resulting in the transconductance plot shown in figure 5.18. The output resistance of the test amplifier was measured to be 2 MΩ. Note that this transconductance plot shows a fairly linear relationship between the transconductance and the control voltage. Although this was an unexpected result, the mechanism behind this linearization is explained in [Babanezhad and Temes, 1985] where a very similar circuit was analyzed. In this analysis, it is shown that if the differential voltage applied to the variable GM stage in figure 5.13 is small, then the transconductance from the input control voltage to the difference in the output currents

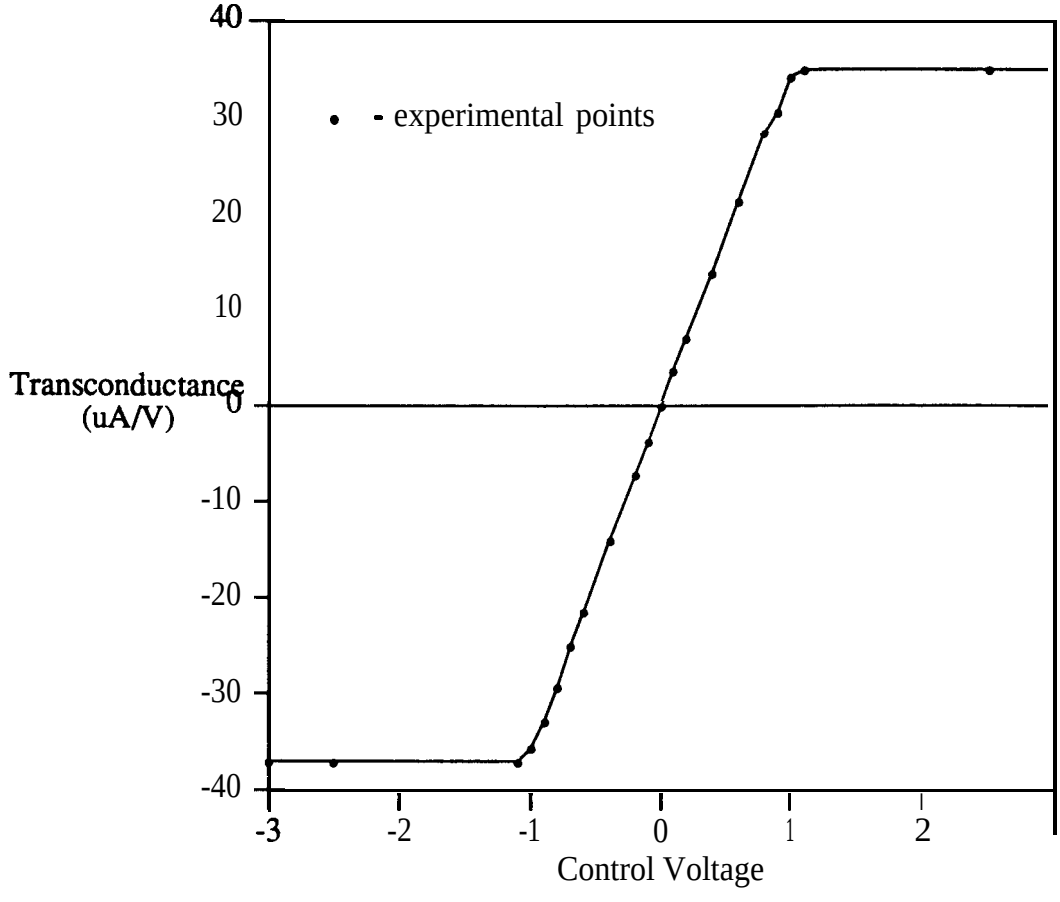


Figure 5.18: Experimental results for variable transconductance amplifier.

approaches a linear relationship. Since in measuring the transconductance, a small differential input voltage signal was used, the linear relationship shown in figure 5.18 was obtained.

Using the transconductance plot in figure 5.18 and arbitrarily choosing 0.4 volts to correspond to a coefficient value of unity, the control voltages necessary to create the lowpass filter described by the system in equation (5.5) were determined. However, note that the programmable filter implements the transposed system of equation (5.5) and forces c_3 to be one. Thus, in fact, the following system is realized.

$$A = \begin{bmatrix} 0 & -0.98361 & 0 \\ 0.98361 & 0 & -1.2307 \\ 0 & 1.2307 & -1.8805 \end{bmatrix} \quad b = \begin{bmatrix} 1.5779 \\ 0 \\ 0.4563 \end{bmatrix} \tag{5.10}$$

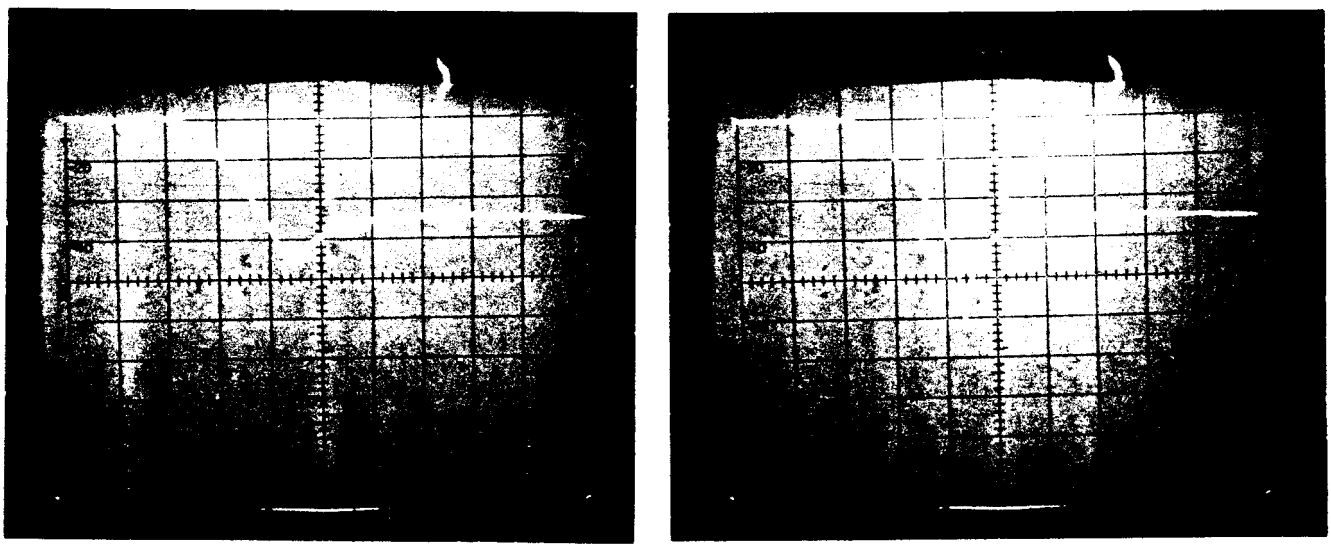
$$c^T = [0 \quad 0 \quad 1] \quad d = 0$$

For this specific example, Alfl, Alf2, and Alf3 were set to 0.394, 0.493 and 0.766 volts, respectively while Bl, B2, and B3 were set to 0.627, 0, and 0.181 volts, respectively. As well, external capacitors of value 2.7 nF were placed on the outputs Xl, X2, and X3 to create the necessary integration so that the passband edge was located at 1 KHz. The experimental spectrum response of the programmable filter for this example is shown in figure 5.19(a) where signal levels were approximately 100 mV peak. Note that although the ideal transfer function has a zero on the imaginary axis, the experimental spectrum response shows that the zero has moved off the axis. It is believed that this effect is due to the finite output impedance of the transconductance amplifiers causing the integrators to be lossy. To demonstrate that this is indeed the case, consider the system in equation (5.10) where the DC gain of the integrators is only 100. For this situation, the system in equation (5.10) is mod&d by subtracting 0.01 off each of the diagonal elements in the A matrix, resulting in the following system

$$A = \begin{bmatrix} -0.01 & -0.98361 & 0 \\ 0.98361 & -0.01 & -1.2307 \\ 0 & 1.2307 & -1.8905 \end{bmatrix} \quad b = \begin{bmatrix} 1.5779 \\ 0 \\ 0.4563 \end{bmatrix} \tag{5.11}$$

$$c^T = [0 \quad 0 \quad 1] \quad d = 0$$

For this lossy integrator system, the poles and zeros are shifted left 0.01 in the s-plane from the equivalent roots in the ideal integrator case and thus the lossy case zeros move off the imaginary axis. To place the poles and zeros back at their desired locations, one could pie-shift the poles and zeros by the amount that they are expected to shift. However, rather than perform this complicated procedure, it was found that by adding a negative term to b_2 the zeros could be shifted right and therefore back on the imaginary axis. Specifically, for the above lossy system, if b_2 is set to -0.0074 the zeros fall on the imaginary axis once again. Although this simple method does



(a) (b)

Figure 5.19: Lowpass transfer function response for programmable filter.
Vertical scale = 10 dB/div Horizontal scale = 500 Hz/div
(a) Response when b_2 is set to zero.
(b) Response after b_2 is adjusted to place zeros on the imaginary axis.

not correct the pole locations, the transfer functions tested in this section appear to be insensitive enough that the final function is close to the desired response. The spectrum response for the programmable filter with b_2 adjusted to create zeros on the imaginary axis is shown in figure 5.19(b) where b_2 was measured to be about -0.05 volts after adjustment.

Finally, to demonstrate that the programmable filter can realize different transfer functions, two spectrum responses for the programmable filter are shown in figure 5.20 where the different responses are created by changing the voltage into the transconductance control inputs. Note

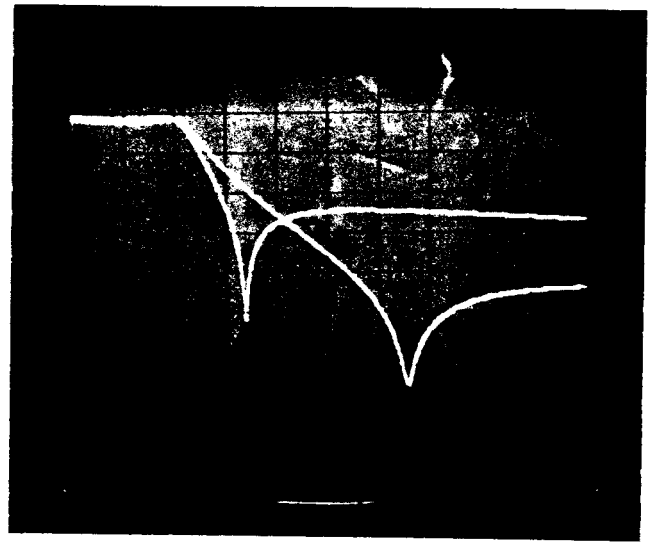


Figure 5.20: Two different lowpass filter responses obtained with the programmable filter.
Vertical scale = 10 dB/div Horizontal scale = 1 KHz/div

that both transfer functions have the same passband yet quite different stopbands and this could only be accomplished by changing both the poles and zeros of the filter. For both of these filters, b_2 was adjusted to place the transmission zeros on the imaginary axis.

5.6. Summary

This chapter has shown experimental and design details for a discrete prototype analog adaptive filter demonstrating that the adaptive algorithms presented in chapter 4 can successfully be transformed into the analog domain. As well, the design details and experimental results for a fabricated monolithic programmable filter were presented. Work is presently being done on adapting the programmable filter and realizing a fully integrated analog adaptive filter.

Chapter 6

The Effects of DC Offsets in Analog Implementations

6.1. Introduction

During experimentation with the discrete prototype analog adaptive IIR filter described in chapter 5, it was found that the system was sensitive to DC offsets present at the integrators used to implement the coefficient update formula. In fact, DC offsets appear to be one of the most severe problems in realizing analog adaptive filters. In the discrete prototype, to overcome the effects of the DC offsets, a large gain was required in the realization of the error signal. Chapter 5 shows the implementation of the large gain on the error signal while this chapter explains how this gain reduces the DC offset effects. Unfortunately, in some applications, realizing this large gain may be difficult to achieve and thus one would like to determine the minimum gain necessary. Thus, it is important to determine the effect of these offsets and develop analytical results that one can use to ensure that practical designs will meet the desired specifications. For these reasons, this chapter investigates the effect of constant offset terms present in the coefficient update formulae. General formulae will be derived giving the excess mean squared error resulting from these offsets for the LMS and sign-data algorithms. This general formula will show that a high correlation between gradient signals increases the excess error due to DC offsets.

Section 6.2 will present a model illustrating the locations of the DC offsets that will be considered in this chapter. In section 6.3, a second-order FIR example will be presented to obtain some insight as to why gradient signals with a high degree of correlation result in a large offset-induced excess error. A general formula for offset-induced excess error will be derived in sec-

tion 6.4 for the case of FIR adaptive filters and in section 6.5, it will be shown that this same formula can be used to give approximate results for the IIR case. Throughout sections 6.2 to 6.5, simulation results using digital adaptive filters will be given to verify the formulae derived, however, to feel confident that the derived formulae are useful in analog implementations, a comparison with experimental results is necessary. To accomplish this comparison, section 6.6 presents a modification of the excess error formula to account for use of the sign-data algorithm that is utilized in the prototype analog adaptive filter. Finally, experimental results are given in section 6.7 showing that the presented formulae agree with results from the discrete prototype. Also presented in section 6.7 is an explanation as to why the large gain on the error signal reduces DC offset effects.

It should be pointed out that the DC offset formulae for the FIR case using the LMS algorithm (the sign-data algorithm was not analyzed) has previously been presented' [Compton, 1988]. However, this previous derivation of the formulae does not easily show that these same formulae apply to the IIR case and thus, a different derivation of these formulae is given in this chapter.

6.2. Coefficient update DC offset modeling

The LMS update formula for the coefficient p_i applied to analog realizations was given in chapter 2 as

$$p_i(t) = 2\mu \int_0^t \left[e(\tau) \frac{\partial y(\tau)}{\partial p_i} \right] d\tau \qquad (6.1)$$

A block diagram of this analog coefficient update formula is shown in figure 6.1. At first glance, this diagram may appear unusual since it appears that the coefficient p_i is obtained as the output

[1] It should be mentioned that this reference was found after the theoretical work in this chapter was performed.

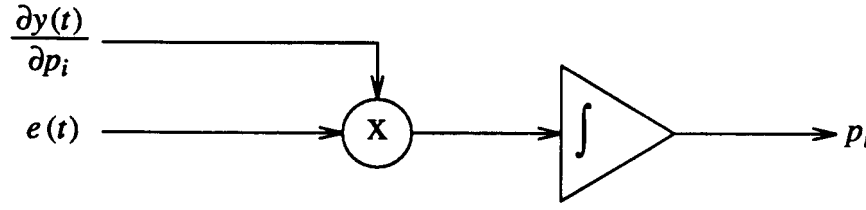


Figure 6.1: Block diagram of coefficient update formula.

of an integrator with no apparent feedback to keep the output signal from saturating. In fact, there is negative feedback present in that the error signal is a function of the parameter p_i (the gradient signal is also a function of p_i but to a lesser extent). At steady state, if the signal p_i starts to drift upwards, the error signal correlates with the gradient signal in such a way as to bring p_i back down. This is the same mechanism which allows the adaptive filter to find a minimum in the performance surface or, equivalently, a location where the error signal is uncorrelated with the gradient signal.

Now consider the effect of a DC offset applied to the integrator used in determining the coefficient p_i. The i'th coefficient update formula with a DC offset becomes

$$p_i(t) = 2\mu \int_0^t \left[e(\tau) \frac{\partial y(\tau)}{\partial p_i} + m_i \right] d\tau \qquad (6.2)$$

where m_i is the DC offset for the i'th update formula. The block diagram for this case is shown in figure 6.2 where the DC offset is injected as a separate signal so that the integrator may be considered ideal.

Note that with the model in figure 6.2, the DC signal need not come from only the DC offset of the integrator but can also include the DC offsets of both the error and gradient signals. It is easily shown that if both the error and gradient signals have DC offsets, then these DC

signals will correlate with each other resulting in an additional DC offset. Therefore, by defining m_i to also include this offset signal, we may consider both the error and gradient signals to be ideal with respect to DC offsets.

When an adaptive filter is at steady state, the expected value of the coefficient signal p_i is a constant value implying that the expected value of the signal into the integrator must be zero. Thus at steady state, the following equation holds

$$E\left[e(t)\frac{\partial y(t)}{\partial p_i} + m_i\right] = 0 \tag{6.3}$$

where $E[\bullet]$ denotes expectation. Since the expected value of a DC signal is the DC level, we can write

$$E\left[e(t)\frac{\partial y(t)}{\partial p_i}\right] = -m_i \tag{6.4}$$

Recall from chapter 2 that the inside of the expectation operator in equation (6.4) is the instantaneous estimate of the derivative of the mean squared error with respect to the parameter p_i, or in mathematical terms,

$$\text{e(f)}\frac{\partial y(t)}{\partial p_i} = \frac{-1}{2}\frac{\partial e^2(t)}{\partial p_i} \tag{6.5}$$

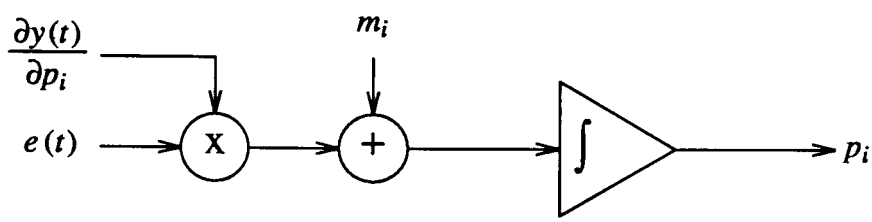


Figure 6.2: Block diagram of coefficient update formula with DC offset m_i.

Substituting equation (6.5) in equation (6.4) above and swapping the expectation and derivative operators, we also have the following condition at steady state.

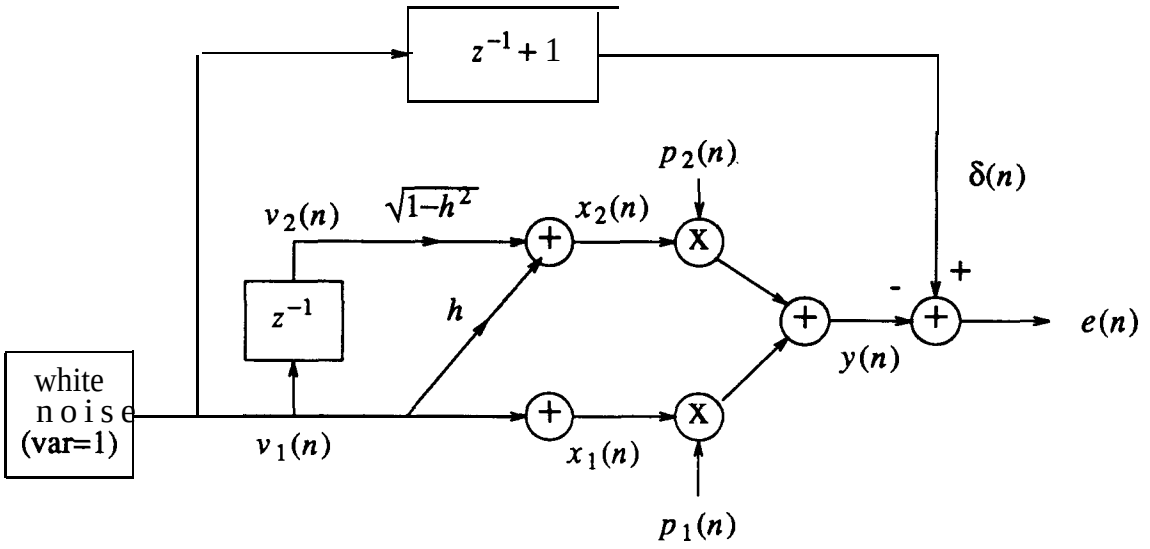
$$\frac{\partial E[e^2(t)]}{\partial p_i} = 2m_i \tag{6.6}$$

This formula implies that when no DC offset is present in the i'th update formula ($m_i = 0$), the adaptive filter settles at a point where the partial derivative of the performance surface with respect to the i'th coefficient is zero. This is precisely the condition for finding a minimum. *However, in the case of a non-zero DC offset, the* adaptive *filter settles at a point where the same* partial derivative is at a value equal to twice the DC offset. In other words, the filter settles at a position where the error is slightly correlated with the gradient signal in order to cancel the effect of the DC offset, as seen from equation (6.4). Note that a DC offset forces the filter coefficients to be incorrect which implies an error in the programmable filter's transfer function at all frequencies (not just at DC).

6.3. Second order example

To obtain some insight into the effects of DC offsets, we will develop a formula giving the excess mean squared error due to offsets for a simple second order FIR example. For this example, we change to the digital domain because of the simplicity of the simulations in that domain with which we can verify our results. This example is designed to illustrate the effect of non-orthogonal gradients on offset-induced error.

The second order example chosen to investigate is shown in figure 6.3. The reference signal is obtained as the output of a simple FIR filter corresponding to adding the present input signal to the previous input signal. Note that the signals $v_1(n)$ and $v_2(n)$ are orthonormal while the signals $x_1(n)$ and $x_2(n)$ both have norms equal to one but are correlated with each other by the factor h. When *h equals zero, the* pair $x_1(n)$ and $x_2(n)$ form an orthonormal set whereas when *h*



$$p_1(n+1) = p_1(n) + 2\mu[e(n)x_1(n)+m_1]$$
$$p_2(n+1) = p_2(n) + 2\mu[e(n)x_2(n)+m_2]$$

Figure 6.3: Second order FIR example for DC offset analysis.

equals one, the same pair are a dependent set. The output of the programmable filter, y(n), is seen to be a weighted sum of the signals $x_1(n)$ and $x_2(n)$ and the update equations for coefficients p_1 and p_2 with DC offsets are found from

$$p_i(n+1) = p_i(n) + 2\mu\left[e(n)x(n) + m_i\right] \tag{6.7}$$

where m_i is the DC offset for the i'th update formula.

From adaptive filter theory, it is easily shown that the performance surface for this example is simply [Widrow and Stearns, 1985]

$$E[e^2] = E[e^2]_{\min} + [\mathbf{p}-\mathbf{p}^*]^T\mathbf{R}[\mathbf{p}-\mathbf{p}^*] \tag{6.8}$$

where $E[e^2]_{\min}$ is the minimum mean squared error (zero for this example), $\mathbf{p}$ is the vector of

coefficients, $\{p_i\}$,

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{6.9}$$

and $\mathbf{p}^*$ is the vector of coefficients corresponding to the minimum mean squared error which for this example is easily found to be

$$\mathbf{p}^* = \begin{bmatrix} 1 - \dfrac{h}{\sqrt{1-h^2}} \\ \dfrac{1}{\sqrt{1-h^2}} \end{bmatrix} \tag{6.10}$$

The input correlation matrix, R, for this example is simply

$$\mathbf{R} = \begin{bmatrix} 1 & h \\ h & 1 \end{bmatrix} \tag{6.11}$$

Setting the minimum mean squared error to zero and making the change of variables,

$$\mathbf{q} = \mathbf{p} - \mathbf{p}^* \tag{6.12}$$

results in the following expression for the mean squared error.

$$E[e^2] = \mathbf{q}^T \mathbf{R} \mathbf{q} \tag{6.13}$$

Expanding this equation results in

$$E[e^2] = q_1^2 + q_2^2 + 2hq_1q_2 \tag{6.14}$$

where q_1 and q_2 are the elements of $\mathbf{q}$.

By setting the mean squared error to constant values, one can immediately see that this example has elliptical contours for the performance surface. In the case where the correlation number, h, is zero, the contours become circles, as expected.

To find the point where the adaptive filter settles, we take the partial derivatives of this mean squared error formula and set them equal to the respective DC offsets as in equation (6.6). Taking these derivatives results in the following equations that must both be satisfied at steady state.

$$q_1 + hq_2 = m_1 \tag{6.15}$$

$$q_2 + hq_1 = m_2 \tag{6.16}$$

Using a graphical approach to solve these two equations, we can plot two lines corresponding to equations (6.15) and (6.16) on the performance surface contour plot. These two lines are lines of constant partial derivative of the error performance surface with respect to the coefficients q_1 and q_2. *The* intersection of the two lines will be the steady state point of the system.

For the case of ***h equal*** to 0.7 and m_1 and m_2 ***both equal*** to zero, figure 6.4 is a contour plot of the performance surface together with the constant partial derivative lines plotted as thick

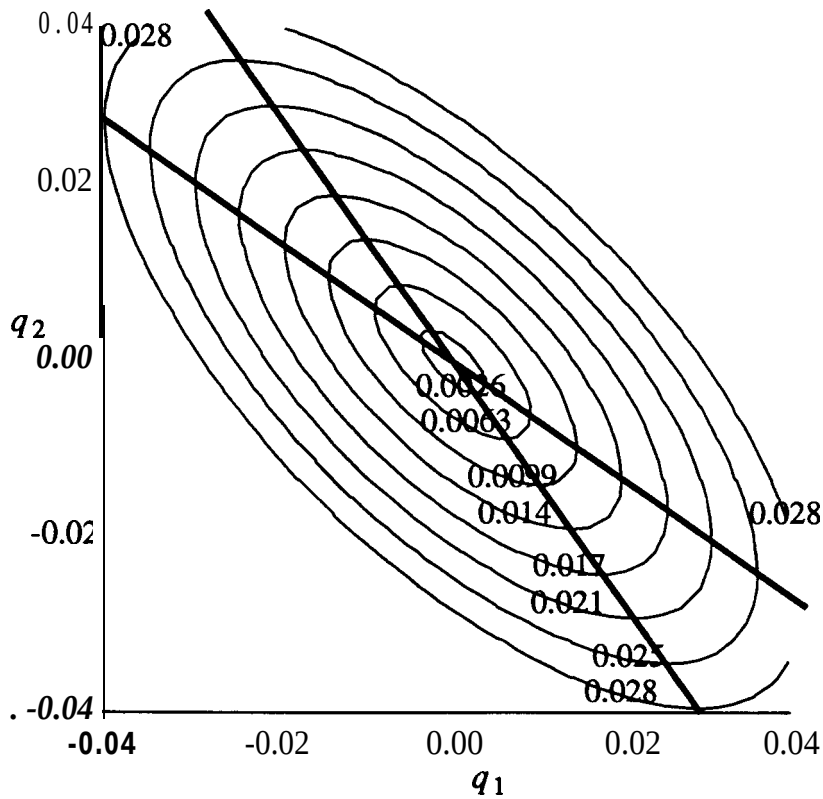


Figure 6.4: Contour plot of performance surface for second order example with $h = 0.7$. Also plotted are the constant derivative lines for $m_1 = m_2 = 0$.

lines. In this case, one sees that the intersection of the two constant partial derivative lines is at the minimum of the performance surface, as expected. As well, note from equations (6.15) and (6.16) that the slopes of these two lines are -*h* and $\frac{-1}{h}$ in this diagram. Thus, as the correlation number *h* approaches unity, the two lines become parallel.

Now consider the case where $m_1 = 0.01$, $m_2 = 0$ and *h* is once again 0.7. Figure 6.5 is a plot of the performance surface and the constant partial derivative lines for this situation. It is seen from figure 6.5 that at steady state, the rms value of the error is certainly not at the minimum in the performance surface. In this case, the steady state rms value of the error is seen

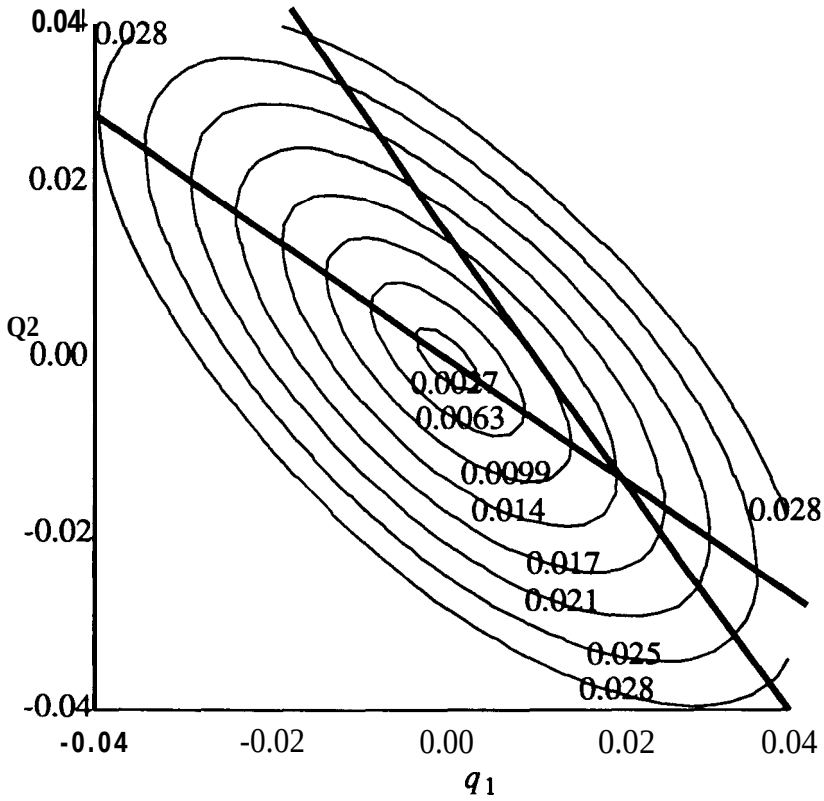


Figure 6.5: Contour plot of performance surface for second order example with $h = 0.7$ Also plotted are the constant derivative lines for $m_1 = 0.01$ and $m_2 = 0$.

to be 0.014. It is interesting to note that as the correlation number, h, approaches unity and the lines become parallel, the excess mean squared error due to offsets will approach infinity.

We can also perform some mathematical manipulations to get a formula giving the rms error for this particular second order example. Solving for the values of q_1 and q_2 using equations (6.15) and (6.16), we have at steady state

$$q_1 = \frac{m_1 - hm_2}{(1-h^2)} \tag{6.17}$$

$$q_2 = \frac{m_2 - hm_1}{(1-h^2)} \tag{6.18}$$

Substituting these values for q_1 and q_2 into the root mean square error formula gives the following result for the excess mean squared error due to DC offsets.

$$|e|_P^2 = \frac{m_1^2 + m_2^2 - 2hm_1m_2}{1-h^2} \tag{6.19}$$

This formula verifies the graphical realization that the excess mean squared error will go to infinity as the states become more correlated.

The above second order example was simulated and root mean squared values estimated by using 5000 samples after steady state was attained. Table 6.1 lists the simulated excess root mean squared error (rmse) for different correlation numbers and DC offsets as well as the calculated rmse. Any differences between the simulated and derived values are believed to be the result of using a finite number of samples to approximate the rms value of a signal as well as using non-zero step sizes to adapt.

6.4. Offset-induced error for the FIR case

In this section, a formula will be derived giving the excess mean squared error due to DC offsets for an N'th order adaptive linear combiner. In the next section, it will be shown that this

m_1	m_2	h	simulated rmse	calculated rmse
0.01	0.0	0	0.010	0.01
0.01	0.0	0.5	0.012	0.0115
0.01	0.0	0.7	0.014	0.0140
0.01	0.0	0.9	0.023	0.0229
0.01	0.0	0.99	0.073	0.0709
0.01	0.02	0.0	0.022	0.0224
0.01	0.02	0.5	0.020	0.0199
0.01	0.02	0.9	0.027	0.0271
0.01	0.02	0.99	0.072	0.0723

Table 6.1: The simulated and calculated root mean squared error (rmse) with varying correlation values, h and DC offsets, m_1 and m_2.

same formula gives approximate results for the general case of an adaptive IIR filter with DC offsets.

Consider the general adaptive linear combiner described in chapter 2 and shown again in figure 6.6. The gradient signals are simply the input states into the linear combiner and can be written as a vector

$$x(n) \equiv \begin{bmatrix} x_1(n) \\ x_2(n) \\ . \\ . \\ x_N(n) \end{bmatrix} \tag{6.20}$$

As well, the correlation matrix, **R,** is defined as

$$\mathbf{R} \equiv E[\mathbf{x}(n)\mathbf{x}^T(n)] \tag{6.21}$$

Referring to figure 6.6 and assuming a small step size, at steady state the coefficients, **p, are** no longer functions of the iteration number and therefore the error signal can be written as

$$\mathrm{e(n)} = \delta(n) - \mathbf{x}^T(n)\mathbf{p} \tag{6.22}$$

Using the same definitions as the last section where $\mathbf{p}^*$ is the vector of coefficients

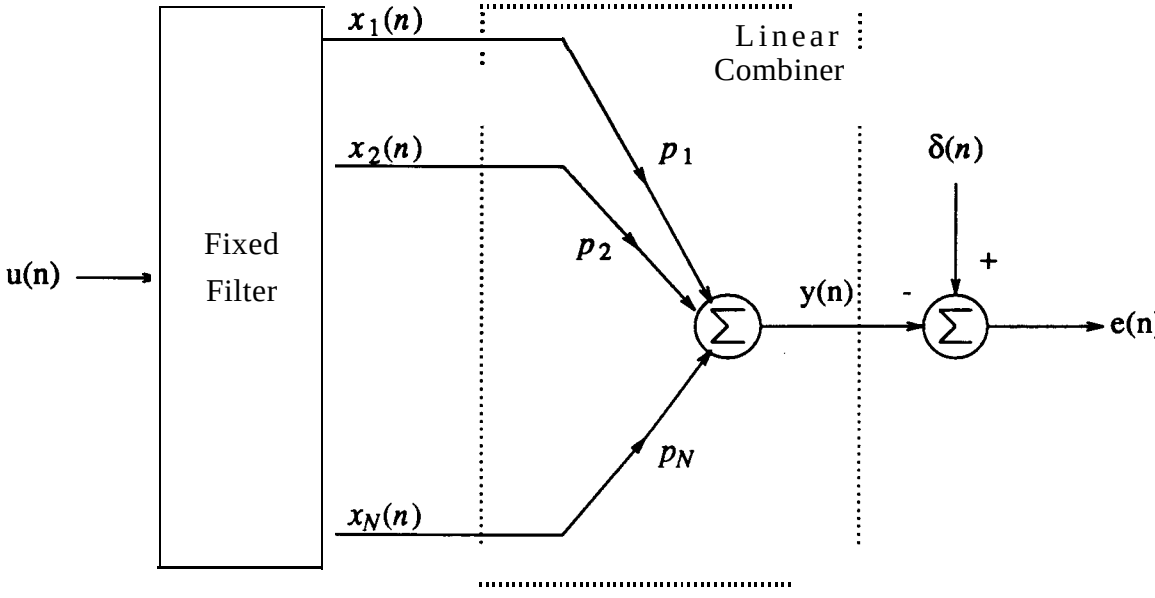


Figure 6.6: A linear combiner adaptive filter

corresponding to the minimum mean squared error and **q** is the difference between **p** and $\mathbf{p}^*$, **we can** write the error as

$$e(n) = \delta(n) - \mathbf{x}^T(n)\mathbf{p}^* - \mathbf{x}^T(n)\mathbf{q} \tag{6.23}$$

Recall that we are interested in finding the excess mean squared error that results from *DC* offsets, therefore we make the assumption that the mean squared error equals zero if all the DC offsets are zero. Making this assumption implies that when $\mathbf{q} = \mathbf{0}$, **the** error signal is always zero and therefore

$$\delta(n) = \mathbf{x}^T(n)\mathbf{p}^* \tag{6.24}$$

Thus, the excess error signal can be reduced to simply

$$e(n) = -\mathbf{x}^T(n)\mathbf{q} \tag{6.25}$$

Now writing the DC offsets as a vector **m, we can** apply equation (6.4) above for the set of DC offsets to obtain

$$E\,[\mathbf{x}(n)e(n)] = -\mathbf{m} \tag{6.26}$$

Combining equations (6.25) and (6.26) results in

$$E\,[\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{q}] = \mathbf{m} \tag{6.27}$$

and using the definition of the correlation matrix,**R, we** obtain the following formula.

$$\mathbf{q} = \mathbf{R}^{-1}\mathbf{m} \tag{6.28}$$

This equation gives the error in coefficient values due to DC offsets. To obtain the excess mean squared error due to DC offsets, we use the definition for the mean squared error and perform the following manipulations,

$$|e|_P^2 = E\,[e(n)e(n)] \tag{6.29}$$

$$= E\,[\mathbf{q}^T\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{q}] \tag{6.30}$$

$$= \mathbf{m}^T\mathbf{R}^{-T}E\,[\mathbf{x}(n)\mathbf{x}^T(n)]\mathbf{R}^{-1}\,\mathbf{m} \tag{6.31}$$

$$= \mathbf{m}^T\mathbf{R}^{-1}\,\mathbf{m} \tag{6.32}$$

Equation (6.32) above is the formula which gives the excess mean squared error due to DC offsets. Note that in the case where all the gradient signals are orthonormal (ie. R equals the identity matrix), the excess mean squared error is simply the sum of the squares of the DC offsets. Also note that the level of the input signal affects the excess error through the correlation matrix **R.** Finally, note that in the FIR case, this correlation matrix, **R,** is not a function of the adaptive coefficients , p_i, since it only depends on the outputs from a fixed filter. For this reason, the offset-induced excess error value obtained through the use of equation (6.32) does not depend on the final transfer function of the adaptive filter. It will be seen that this is not the case for the IIR situation to be described in the next section.

As a test for this error formula, one can apply it to the second order example described in the last section and obtain the same result that was derived there. As an additional test, a third order adaptive linear combiner with DC offsets was simulated where the reference transfer function was $z^{-2} + z^{-1} + 1$. The input correlation matrix **R** for the simulation was chosen to be

$$\mathbf{R} = \begin{bmatrix} 1 & 0.8 & 0.9 \\ 0.8 & 1 & 0.72 \\ 0.9 & 0.72 & 1 \end{bmatrix} \tag{6.33}$$

and the DC offset vector **m** was arbitrarily set to

$$\mathbf{m} = \begin{bmatrix} 0.01 \\ 0.02 \\ -0.03 \end{bmatrix} \tag{6.34}$$

For this example, the calculated excess rmse using equation (6.32) is 0.0922 and the simulated rmse was 0.092.

6.5. Offset-induced error for the IIR case

Note that equation (6.32) was derived for adaptive linear combiners and does not strictly apply in the case of adaptive IIR filters. Fortunately, by redefining the gradient vector x to include all gradients for the IIR case and assuming the DC offsets cause small changes in the coefficients $\{p_i\}$, equation (6.32) can be used to approximate the excess mean squared error. Specifically, the elements of the gradient vector, $\mathbf{x}(n)$, for the IIR case are defined as

$$x_i(n) = \frac{\partial y(n)}{\partial p_i} \tag{6.35}$$

The correlation matrix, **R,** is defined, as before, to be

$$\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)] \tag{6.36}$$

Note, that in the case where white noise is applied to the system input, the correlation matrix **R** can be obtained by using impulse responses as described in chapter 2. Specifically, the element R_{ij} can be found from the following equation

$$R_{ij} = \sum_{n=0}^{\infty} \frac{\partial y(n)}{\partial p_i} \frac{\partial y(n)}{\partial p_j} \tag{6.37}$$

where $\frac{\partial y(n)}{\partial p_i}$ and $\frac{\partial y(n)}{p_j}$ are the gradient responses when an impulse is applied to the system input and the coefficients are fixed at their final values. If continuous-time rather than discrete-time circuits are being used then the summation function in equation (6.37) above is changed to an integration. Finally, the DC offset vector, **m,** is defined, as before, to be the DC offsets introduced in each coefficient update formula.

To illustrate that this equation applies to the IIR case, one can follow the same analysis as in the adaptive linear combiner situation except that writing the error as equation (6.25) must be justified. This equation can be justified, if small coefficient changes are assumed since with this assumption, the error can be written as

$$e(n) = \delta(n) - y^*(n) - \sum_{i=1}^{N} \frac{\partial y(n)}{\partial p_i} \Delta p_i \tag{6.38}$$

where y*(n) is defined as the output $y(n)$ obtained with the optimum coefficients causing minimum mean squared error and Δp_i is defined to be the change in coefficients due to DC offsets. As before, since we are interested in excess MSE due to DC offsets, we can assume the minimum MSE is zero and write

$$e(n) = \mathbf{x}^T(n)\mathbf{q} \tag{6.39}$$

where the elements of **q are** Δp_i**.** This small coefficient change assumption justifies the use of equation (6.39) above and therefore one can use equation (6.32) to approximate the excess error due to DC offsets in adaptive IIR filters.

Note that in the IIR case, the correlation matrix, R, is a function of the adapting coefficients, p_i, and is therefore a function of the adaptive filter's transfer function. This implies that in the IIR case, one requires knowledge of the final transfer function for the adaptive filter in

order to apply the offset-induced excess error formula. Since this exact transfer function is usually not known, one can only hope to obtain approximate results with this method by estimating the final transfer function.

To check the validity of this formula for the IIR case, a second order model matching example with DC offsets present was simulated. In this simulation, only the bottom row of the A matrix was adapted while all other parameters remained equal to the optimum values. The reference filter had the state-space describing equations:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -0.8 & 1.7 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0.125 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad d = 0 \tag{6.40}$$

Defining $\mathbf{x}(n)$ to be the vector of gradient signals required to adapt the bottom row of A, and assuming the final transfer function of the adaptive filter equals the transfer function described in equation (6.40), for the white noise input case, the correlation matrix, R, was found to be

$$\mathrm{R} = \begin{bmatrix} 1.032 & 0.9913 \\ 0.9913 & 1.032 \end{bmatrix} \tag{6.41}$$

With offsets equal to 0.01, the simulated and calculated rms errors were 0.1 and 0.07, respectively. The bottom row of the A matrix settled at the coefficient values -0.75 and 1.65. This example shows a reasonable agreement between the calculated and simulated values. However, by decreasing the offset, the accuracy of the small change approximation in equation (6.38) is improved and therefore an even closer agreement should be obtained. Decreasing the offsets to 0.001, the simulated and calculated rms errors were 0.007 and 0.00704, respectively, which is certainly a close agreement. For this simulation, the bottom row of the A matrix settled at the coefficient values -0.797 and 1.697.

Finally, note from equation (6.32) that the value of the excess error due to DC offsets is proportional to the inverse of the correlation matrix, R. This fact implies that the excess error will increase as the states become more con-elated since the matrix R will become more ill-

conditioned. This increased excess error is another reason to look for structures with orthonormal gradients.

6.6. Offset-induced error for the sign-data algorithm

Recall from chapter 5 that the sign-data algorithm mutiplies the error signal by the sign of the gradient signal rather than the gradient signal itself for reduced hardware complexity. Therefore, for the many practical applications that use the sign-data algorithm, the offset-induced excess error formula needs be adjusted to account for this different adaptive algorithm.

In accounting for the use of the sign-data algorithm, we make use of the signum function, $sgn[x]$, defined in the following equation.

$$sgn[x] = 1 \text{ if } x \geq 0 \tag{6.42}$$
$$= -1 \text{ if } x < 0$$

With this signum function, equation (6.26), above, is replaced by

$$E[sgn[\mathbf{x}(n)]e(n)] = -\mathbf{m} \tag{6.43}$$

where the signum of a vector is defined as applying the signum function to each of the vector elements. As before, we can write the error signal as a function of $\mathbf{q}$ and $\mathbf{x}(n)$ using equation (6.25) and therefore can write

$$E[sgn[\mathbf{x}(n)]\mathbf{x}^T(n)]\mathbf{q} = \mathbf{m} \tag{6.44}$$

This leads to the following equation for $\mathbf{q}$.

$$\mathbf{q} = \tilde{\mathbf{R}}^{-1}\mathbf{m} \tag{6.45}$$

where the signum correlation matrix, $\tilde{\mathbf{R}}$, is defined as

$$\tilde{\mathbf{R}} = E[sgn[\mathbf{x}(n)]\mathbf{x}^T(n)] \tag{6.46}$$

or, equivalently, the element R_{ij} is defined as

$$\tilde{R}_{ij} = E[sgn[x_i(n)]x_j(n)] \tag{6.47}$$

Now, we can use equation (6.30) and substitute in the new expression for $\mathbf{q}$ to obtain the following offset-induced excess error formula for the sign-data case.

$$\|e\|_p^2 = \mathbf{m}^T \tilde{\mathbf{R}}^{-T} \mathbf{R} \tilde{\mathbf{R}}^{-1} \mathbf{m} \tag{6.48}$$

Note, that in the above equation, the matrices R and $\tilde{\mathbf{R}}$ need to be obtained. For white noise inputs, the matrix R can be obtained as has been shown above. Unfortunately, it is not clear how to obtain the matrix $\tilde{\mathbf{R}}$ except by applying the defining equation (6.47). However, 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 $\tilde{\mathbf{R}}$ in terms of the elements of R. Specifically, if the input signal has a zero mean Gaussian distribution, the joint probability density function, $\Phi_{x_i x_j}(x_i, x_j)$, between the signals x_i and x_j can be written as [Papoulis, 1984, p. 186]

$$\Phi_{x_i x_j}(x_i, x_j) = \left[\frac{1}{4\pi^2 [R_{ii} R_{jj} - R_{ij}^2]} \right]^{1/2} \exp\left[\frac{-1}{2} [x_1 \; x_2] \begin{bmatrix} R_{ii} & R_{ij} \\ R_{ij} & R_{jj} \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right] \tag{6.49}$$

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

$$\tilde{R}_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} sgn[x_1] x_2 \Phi_{x_i x_j}(x_i, x_j) dx_1 dx_2 \tag{6.50}$$

Performing this integration (which is not a trivial process!) leads to the following closed form expression for the elements of $\tilde{\mathbf{R}}$.

$$\tilde{R}_{ij} = \left[\frac{2R_{ij}^2}{\pi R_{ii}} \right]^{1/2} \tag{6.51}$$

Therefore, in the case of zero mean Gaussian white noise inputs, equation (6.51) can be used to obtain $\tilde{\mathbf{R}}$ and equation (6.48) can be used to give the offset-induced excess error in adaptive FIR filters using the sign-data algorithm. The same formulae can be used to give approximate results for adaptive IIR filters also using the sign-data algorithm.

6.7. Experimental Results

In this section, DC offset experimental results using the discrete prototype will be compared to the theoretical results using the offset-induced excess error formulae above. Since the discrete prototype uses the sign-data algorithm, equation (6.48) will be the formula used for comparison.

Figure 6.7 shows the method of applying a known DC offset signal to the i'th coefficient update integrator. First, note in figure 6.7, that the sign of the gradient signal is multiplied by $ke(t)$ where k is an amplification constant for the error signal to reduce the offset effects. In chapter 5, it was seen that k 'was arbitrarily chosen to be 82 for the discrete prototype. It is easily seen that this reduces the offset effects since adding this gain factor implies that $\|ke\|_P^2$ will replace $\|e\|_P^2$ in equation (6.48) above. Since, the right hand side of equation (6.48) is unaffected by the addition of the gain factor, k, *the* resulting offset-induced excess rms error is reduced by the factor k. Experimentation confirms the reduction in offset-induced excess error when increasing the gain factor, k. It should be pointed out that this gain factor will be difficult to realize at high frequencies since high frequency gain circuits are not a trivial task to implement. This difficulty in implementation is one of the major reasons for developing these DC offset related formulae. With these formulae 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 figure 6.7, an equivalent DC offset voltage was applied to the i'th coefficient update integrator by adding the resistor network shown in figure 6.7(a) to the virtual ground terminal. This method of applying the DC offset was chosen so that the connections to the discrete prototype could be simply added on in parallel rather than having to "cut" into the circuit. As shown in figure 6.7(b), the magnitude of the applied equivalent DC offset was

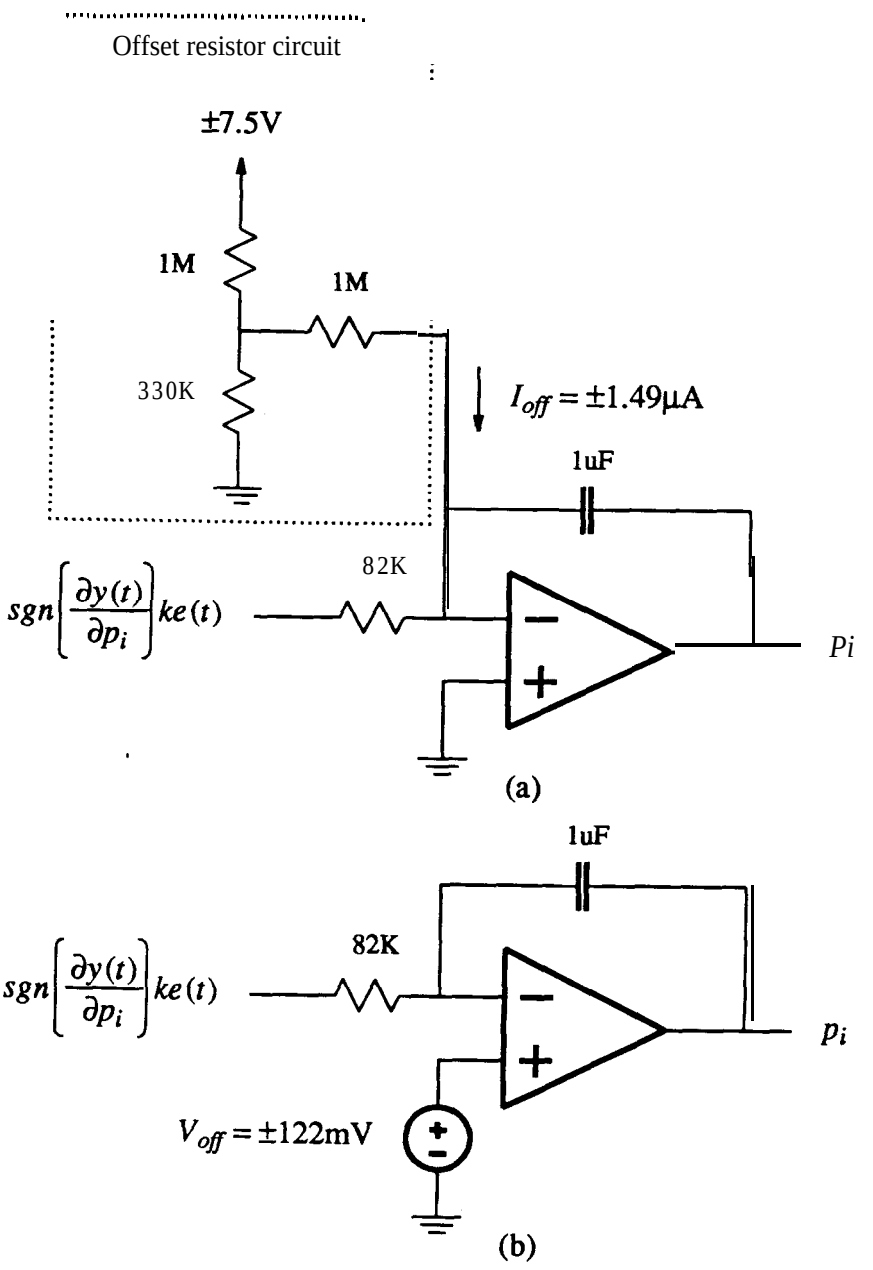


Figure 6.7: Injecting a DC offset into the coefficient update integrator.
(a) Experimental method
(b) Equivalent circuit

122mV

(an arbitrary number dependent on resistor values) where the sign of the offset is determined by the sign of the supply voltage that is applied to the offset resistor network. With this approach, two offset-induced excess rms error voltages are measured, $\|ke\|_p$ and $\|ke\|_n$, which correspond to using a positive and negative voltage supply, respectively. If no other offset voltages are present in the circuit, then the above theory predicts that $\|ke\|_p$ should equal $\|ke\|_n$. However, this is not observed due to the fact that there already exists unknown equivalent offsets resulting from the non-idealities of the circuit realization. If we call the vector of unknown offsets $\mathbf{m}_1$ and the measured rms error with no external offsets applied, $\|ke\|_{\mathbf{m}_1}$, then we can write

$$\|ke\|^2_{\mathbf{m}_1} = \mathbf{m}_1^T \mathbf{H} \mathbf{m}_1 \tag{6.52}$$

where $\mathbf{H}$ is defined to be $\tilde{\mathbf{R}}^{-T}\mathbf{R}\tilde{\mathbf{R}}^{-1}$. Now letting the known vector of positive applied offsets be $\mathbf{m}_2$, we can write equations for $\|ke\|_p$ and $\|ke\|_n$.

$$\|ke\|_p^2 = (\mathbf{m}_1 + \mathbf{m}_2)^T \mathbf{H} (\mathbf{m}_1 + \mathbf{m}_2) \tag{6.53}$$

$$= \mathbf{m}_1^T \mathbf{H} \mathbf{m}_1 + \mathbf{m}_2^T \mathbf{H} \mathbf{m}_2 + \mathbf{m}_1^T \mathbf{H} \mathbf{m}_2 + \mathbf{m}_2^T \mathbf{H} \mathbf{m}_1 \tag{6.54}$$

and

$$\|ke\|_n^2 = (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{H} (\mathbf{m}_1 - \mathbf{m}_2) \tag{6.55}$$

$$= \mathbf{m}_1^T \mathbf{H} \mathbf{m}_1 + \mathbf{m}_2^T \mathbf{H} \mathbf{m}_2 - \mathbf{m}_1^T \mathbf{H} \mathbf{m}_2 - \mathbf{m}_2^T \mathbf{H} \mathbf{m}_1 \tag{6.56}$$

Letting $\|ke\|_{\mathbf{m}_2}$ be the offset-induced excess error due to $\mathbf{m}_2$ only, from equations (6.52), (6.54) and (6.56), it is not difficult to show the following equality.

$$\|ke\|^2_{\mathbf{m}_2} = \tfrac{1}{2}\|ke\|_p^2 + \tfrac{1}{2}\|ke\|_n^2 - \|ke\|^2_{\mathbf{m}_1} \tag{6.57}$$

Finally, we can now compare experimental results with theoretical ones. For the DC offset experimentation, the reference filter was set to the same as that used in the first experimental results presented in chapter 5. Specifically, the reference filter corresponded to equation (5.5) and the adaptive filter cotresponded to equation (5.4). For this case, the $\mathbf{R}$ and $\tilde{\mathbf{R}}$ matrices are found to be:

$$\mathbf{R} = \begin{bmatrix} -1.0000 & 0 & 0 & 0.2483 & 0.4986 & 0.3246 \\ 0 & 1.0000 & 0 & -0.4986 & -0.1579 & 0.0032 \\ 0 & 0 & 1.0000 & 0.3246 & -0.0032 & 0.0966 \\ 0.2483 & -0.4986 & 0.3246 & 0.8187 & 0 & 0.2171 \\ 0.4986 & -0.1579 & -0.0032 & 0 & 0.5470 & 0 \\ 0.3246 & 0.0032 & 0.0966 & 0.2171 & 0 & 0.2284 \end{bmatrix} \quad (6.58)$$

$$\tilde{\mathbf{R}} = \begin{bmatrix} -0.7979 & 0 & 0 & 0.1981 & 0.3978 & 0.259 \\ 0 & 0.7979 & 0 & -0.3978 & -0.1260 & 0.0026 \\ 0 & 0 & 0.7979 & 0.259 & -0.0026 & 0.077\,1 \\ 0.2189 & -0.4397 & 0.2863 & 0.7219 & 0 & 0.1915 \\ 0.5379 & -0.1704 & -0.0035 & 0 & 0.5901 & 0 \\ 0.5420 & 0.0053 & 0.1614 & 0.3625 & 0 & 0.3813 \end{bmatrix} \quad (6.59)$$

Table 6.2 shows a comparison of theoretical vs. experimental excess error voltages due to DC offsets. For this comparison, white noise was applied to the system input and with no applied offsets, the rms voltage of the error voltage was measured giving $|ke|_{m_1}$. Then, a positive and negative offset voltage was applied to the i'th coefficient integrator so that $|ke|_p$ and $|ke|_n$ could be measured. With these three measurements, the experimental value for $|ke|_{m_2}$ can be derived using equation (6.57). Finally, this experimental value of $|ke|_{m_2}$ is compared against the

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

Table 6.2: A comparison of theoretical and experimental results for injected DC offsets. When no DC offsets applied, $|ke|_{m_1}$ equals 0.2 Vrms. The i'th row corresponds to DC offsets on the i'th integrator.

theoretical value obtained using equation (6.48). This corresponds to each row in table 6.2 relating to a DC offset vector of $\pm 0.122\mathbf{v}_i$ applied where $\mathbf{v}_i$ is a basis vector with a value of unity in the i'th row. To measure the rms voltages of the amplified error, $|ke|$, a digital-readout true rms meter was used. Unfortunately, it was not a simple matter to read the rms value of the output error signal since low frequency components were present and thus successive meter readings varied considerably. The value used was an estimate of the average of a few successive readings. Note that $|ke|_{m_1}$ was measured to be 0.2 Vrms. Also, note that in table 6.2, the same notation for the adapting coefficients is used as that in equation (5.4). We see from table 6.2 that all measurements agree within 20 percent of the theoretical predictions. This degree of accuracy is reasonable considering that all circuit non-idealities other than integrator DC offsets have been ignored and that noise rms measurements were made. This degree of accuracy should be close enough for design purposes when one considers the variability of DC offsets in a given technology.

6.8. Summary

Through the use of experimentation with a discrete prototype, it was observed that DC offsets on the coefficient integrators appears to be the most severe non-ideal effect of analog adaptive filters. To reduce the effect of these DC offsets, a high gain was used to amplify the error signal. (A explanation of how this gain reduces DC offsets was presented in section 6.7.) However, at high frequencies, this gain may be difficult to implement and thus, a designer needs some guidance in choosing a minimum gain that guarantees that the system will meet specifications.

Towards choosing this minimum gain, this chapter has developed formulae which give the expected excess error and the coefficient deviations due to the DC offsets of these integrators. It

was shown that the derived formulae are exact for the adaptive linear combiner case (FIR), whereas for the IIR case, the formulae are an approximation using first order sensitivities. These formulae were also modified for the sign-data LMS algorithm so that realizations using this algorithm could be analyzed. Finally, experimental results were compared to theoretical values showing a close agreement between the two sets.

Chapter 7

Summary and Conclusions

7.1. Introduction

The main purpose of this thesis was to investigate the feasibility of analog adaptive recursive filters. The author believes that using the approach developed in this thesis, analog recursive filters arc certainly feasible. However, along the way of this feasibility investigation, ideas have been developed which should be useful in more areas than just analog adaptive recursive filters. In particular, the orthonormal ladder filter appears to be a viable alternate structure to a cascade of biquads. As well, the adaptive algorithms presented in chapter 4 seem to be of some use in the digital domain.

Section 7.2 will summarize the material presented in this thesis and outline the main contributions made. Suggestions for further research will be presented in section 7.3.

7.2. Summary

In chapter 2, necessary background material was presented including notation usage and the definition of some commonly used terms; expectation, correlation and norms. As well, some adaptive filter theory was presented where it was shown that orthonormal signals are useful when using the LMS adaptive algorithm. Also presented in this chapter was a brief introduction to state-space filter theory. Of particular importance were the definitions of the correlation matrices K and W and the idea of a transposed system such that the intermediate-functions, $\mathbf{F}(s)$ and $\mathbf{G}(s)$, are exchanged.

The first contribution made in this thesis was the presentation in chapter 3 of a new filter structure resulting in circuits referred to as "orthonormal ladder filters". Through the use of examples, it was shown that orthonormal ladder filter realizations have a sensitivity and dynamic range performance comparable to a cascade of biquads. However, other interesting properties make this new filter structure useful in the design of adaptive filters. Specifically, it was shown that the inherent structure of orthonormal ladder filters guarantees that the resulting realizations are L_2 scaled for optimum dynamic range. As well, it was shown that the set of integrator outputs in these filters form an orthonormal set when white noise is applied at the filter input. Finally, it was shown that the sign of only one coefficient determines the stability of the system and thus provides a simple stability check.

The next main contribution was presented in chapter 4 where new adaptive algorithms for state-space recursive systems were given which could be applied in either the digital or analog domain. A general algorithm for adapting all the coefficients of a state-space system was first presented but, unfortunately, requires an excessive amount of computations. To reduce the amount of computations, single-row and single-column adaptive structures were presented. Although these adaptive filters have no restrictions on their pole locations, it is felt that these new filters are best suited to applications where an estimate of final pole locations is known. For these types of applications, it was shown that these new adaptive filters have significant performance improvements over the traditional direct-form implementations, especially in the practical case of oversampled systems. It should be pointed out that the adaptive algorithms presented in this thesis are all based on the LMS steepest descent approach and would have to be modified to ensure global rather than local convergence.

In chapter 5, it was demonstrated that the algorithms of chapter 4 could successfully be converted to the analog domain and that fully integrated analog adaptive recursive filters are possible. Towards demonstrating the successful conversion of the adaptive algorithms, design details and experimental results were presented for a discrete prototype of a third-order single-row analog adaptive filter. The results are very encouraging. With respect to fully integrating an analog adaptive recursive system, design details and experimental results were given for a monolithic CMOS programmable filter. The results for this programmable filter show that only a small amount of silicon is required and that the programmable filter's performance indicates good programmability. Thus, though an extra filter is required to create gradient signals to adapt the programmable filter, it should be possible to fully integrate an analog adaptive recursive filter.

Finally, chapter 6 addressed the important issue of the effects of DC offsets present in analog adaptive recursive filter realizations. Formulae giving the excess error and coefficient deviation due to DC offsets were developed for both the LMS and sign-data algorithm for both FIR and IIR filters. The formulae for the LMS case were verified using simulations, and experimental results showed a close agreement between predicted and measured results for the sign-data case. As well, it was shown that by increasing the gain of the error signal, the effects of DC offsets could be reduced.

7.3. Suggestions for further work

The orthonormal filter structure developed in chapter 3 has non-zero elements in each row and each column. Therefore, to continuously adapt a programmable filter with this structure, one requires N extra gradient filters where N is the order of the programmable filter. One way to reduce this computational burden would be to find an orthonormal structure with arbitrary pole

locations but with the varying elements all in only one column and one row. If such a structure could be found, then one would require only 2 extra gradient filters for an N'th order filter. The advantage of such a structure might be that performance improvements similar to single row and single column adaptive filter would be attained without the need for final pole location estimates. The reason for believing that such an improvement would be obtained is that this type of improvement was observed during adaptive filter simulations with the orthonormal structure where N extra gradient filters were used to obtain gradients. Of course, it is entirely possible that such a column and row structure does not exist.

Another area of research to investigate is to modify the Lyapunov formula given in chapter 3 to account for input signals with non-white statistics and thus obtain orthonormal structures for arbitrary inputs. This would be useful where the input signal's statistics can be estimated and they vary significantly from white noise. Along the same thought would be to look for some sort of self-orthogonalizing structure for analog circuits similar to digital adaptive lattice structures.

With regard to the adaptive algorithms presented in chapter 4, it would be very useful if these algorithms could be modified to ensure global convergence. Although, this may be quite a theoretical challenge, the final algorithm could be quite simple. For example, the SHARF algorithm [Larimore et al, 1980] is different from the approximate gradient approach [Feintuch, 1976] in that the SHARF algorithm merely requires a filter on the error signal and the a strictly positive real condition must be satisfied.

With respect to analog implementations, the effects of adapting coefficients clipping should be investigated. This clipping effect could cause local minima to be created depending on the performance surface.

Also on a theoretical level, much work could be done to improve the adaptation speed of convergence such as using different step size constants for different coefficients and varying the step size during adaptation.

With respect to the more practical aspects of future research, the next logical step in this research is to adapt the programmable IC filter using external circuitry and then build a fully integrated analog adaptive recursive filter. Since the structure of the programmable filter is that of an orthonormal ladder filter and only one gradient filter is available, the gradient filter will have to be multiplexed to adapt a single column at a time. Although this will reduce the adaptation speed, the system should perform well without an estimate of final pole locations. As well, a fully integrated single row or column adaptive filter should be constructed with some application in mind.

Note that since all gradients are obtained as outputs of filters using a state-space system description, one could construct adaptive filters with arbitrary operators, for example, damped integrators. This would involve replacing the "s" or "z" operator with some other type of operator. In particular, operators in switched-capacitor filters could be used to adapt SC filters. It would be interesting to see if this type of approach led to any advantages.

Along the lines of the DC offset results, more effort could be applied to reducing the effects of DC offsets without the need for such a high gain on the error signal.

Of course, the analog adaptive recursive filter approach developed here should be applied to some practical applications.

Finally, it should be pointed out that adapting continuous-time integrated filters to match desired transfer functions appears to be quite similar to the problem of self-tuning “fixed” continuous-time integrated filters [Tsividis et al, 1986]. With this in mind, interesting research

might involve applying many of the analog adaptive concepts in this thesis toward the implementation of high quality continuous-time integrated filters.

References

[Amit and Shaked, 1988]
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.

[Ayala, 1982]
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.

[Babanezhad and Temes, 1985]
J.N. Babanezhad and G.C. Temes, "A 20-V Four Quadrant CMOS Analog Multiplier", *IEEE J. Solid-State Circuits,* vol. *SC-20,* pp. 1158-1168, Dec. 1985.

[Banu and Tsividis, 1983]
M. Banu and Y. Tsividis, "Fully Integrated Active RC Filters in MOS Technology", *IEEE J. Solid-State Circuits,* vol. SC-18, pp. 664-651, Dec. 1983.

[Brockett, 1970]
R.W. Brockett, *Finite Dimensional Linear Systems, New* York, N.Y.: John Wiley and Sons, 1970.

[Chen, 1984]
Chi-Tsong Chen, *Linear System Theory and Design, New* York, N.Y.: Holt Rinehart and Winston, 1984.

[Compton, 1988]
R.T. Compton, JR., *Adaptive Antennas,* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1988.

[Eriksson and Allie, 1988]
L.J. Eriksson and M.C. Allie, "System Considerations for Adaptive Modelling Applied to Active Noise Control", 1988 *IEEE International Symposium on Circuits and Systems,* pp. 2387-2390, Espoo, Finland, June 1988.

[Fan and Jenkins, 1986]
H. Fan and W.K. Jenkins "A New Adaptive IIR Filter", *IEEE Trans. on Circuits and Systems,* vol. CAS-33, pp. 939-947, Oct. 1986.

[Feintuch, 1976]
P.L. Feintuch, "An Adaptive Recursive LMS Filter", *Proc IEEE,* vol. 64, pp. 1622-1624, Nov. 1976.

[Hsia, 198 1]
T.C. Hsia, "A Simplified Adaptive Recursive Filter Design", *Proc IEEE,* vol. 69, pp. 1153-1155, Sept. 1981.

[Humpherys, 1970]
D.S. Humpherys, *The Analysis, Design, and Synthesis of Electrical Filters,* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1970.

[Khorramabadi and Gray, 1984]
H. Khorramabadi and P.R. Gray, 'High-frequency CMOS continuous-time filters", *IEEE J. Solid-State Circuits,* vol. SC-19, pp. 939-948, Dec. 1984.

[Jackson, 1970]
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.

[Johns et al., 1987]
D.A. Johns, W.M. Snelgrove and A.S. Sedra, 'Orthogonal Filters and Singly-Terminated LC Ladder Filters", *30'th Midwest Symposium on Circuits and Systerns,* pp. 761-764, Syracuse, New York, Aug. 1987.

[Johnson and Larimore, 1977]
C.R. Johnson and M.G. Larimore, "Comments on and additions to 'An adaptive recursive LMS filter"', *Proc IEEE, vol. 65,* pp. 1399-1402, Sept. 1977.

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

[Lev-Ari et al., 1987]
H. Lev-Ari, J.M. Cioffi and T. Kailath, "Continuous-Time Least Squares Fast Transversal Filters," *Proc. 1987 IEEE Int. Conf. Acoust. Speech and Signal Processing,* pp. 415-418, Dallas, Texas, April 1987.

[Larimore et al., 1980]
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.

[Lee, 1960]
Yuk Wing Lee, *Statistical Theory of Communications, New* York, Wiley, 1960.

[Martin and Sun, 1986]
K.W. Martin and M.T. Sun, "Adaptive Filters Suitable for Real-Time Spectral Analysis", *IEEE Trans. on Circuits and Systems,* vol. *CAS-33,* pp. 218-229, Feb. 1986.

[Mikhael and Yassa, 1982]
W.B. Mikhael and F.F. Yassa, "Stable HighOrder, Continuous Adaptive Filters," *IEEE Int. Symp. on Circuits & Systems Proc.,* pp. 666-669, Rome, Italy, May 1982.

[Moschytz, 1975]
G.S. Moschytz, *Linear Integrated Networks: Design. New* York, Van Nostrand Reinhold, 1975.

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

[Nedungadi and Viswanathan, 1984]
A. Negungadi and T.R. Viswanathan, "Design of Linear CMOS Transconductance Elements", *IEEE Trans. on Circuits and Systems,* vol. CAS-31, pp. 891-894, 1984.

[Gppenheim and Schafer, 1975]
A.V. Gppenheim and R.W. Schafer, *Digital Signal Processing,* Englewood Cliffs, New Jersey, Prentice-Hall, Inc., 1975.

[Papoulis, 1984]
Athanasios Papoulis, *Probability, Random Variables and Stochastic Processes, 2nd ed., New* York, New York, McGraw-Hill Inc., 1984.

[Parikh et al., 1980]
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.

[Qureshi, 1985]
S.U.H. Qureshi, "Adaptive Equalization," ***Proc IEEE,*** vol. ***73, No. 9,*** pp. 1349-1387, Sept. 1985.

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

[Schoeffler, 1964]
J.D. Schoeffler, "The Synthesis of Minimum Sensitivity Networks", ***IEEE Trans. Circuit Theory,*** vol. CT-l 1, pp. 27 1-276, 1964.

[Sedra and Brackett, 1978]
A.S. Sedra and P.O. Brackett, ***Filter Theory and Design: Active and Passive.*** Portland, Oregon, Matrix, 1978.

[Seevinck and Wassenaar]
E. Seevinck and R. F. Wassenaar, "A Versatile CMOS Linear Transconductor/Square-Law Function Circuit", ***IEEE Journal of Solid-State Circuits, vol. SC-22,*** pp. 366-377, June 1987.

[Snelgrove, 1982]
W.M. Snelgrove, "Intermediate-Function Synthesis", Ph.D. dissertation, Univ. of Toronto, Toronto, Ont., Canada, 1982.

[Snelgrove and Sedra, 1986]
W.M. Snelgrove and A.S. Sedra, "Synthesis and Analysis of State-Space Active Filters Using Intermediate Transfer Functions", ***IEEE Trans. on Circuits and Systems, vol. CAS-33,*** pp. 287-301, March 1986.

[Stearns et al., 1976]
S.D. Stearns, G.R. Elliott and N. Ahmed, "On Adaptive Recursive Filtering", *Proc. 10'th Asilomar Conf. Circuits Syst. Cornput.,* pp. 5-10, Nov. 1976.

[Stearns, 1981]
S.D. Stearns, "Error Surfaces of Recursive Adaptive Filters", *IEEE Trans. on Circuits and Systems,* vol. *CAS-28,* pp. 603-606, June 1981.

[Treichler et al., 1987]
J.R. Treichler, C.R. Johnson and M.G. Larimore, *Theory and Design of Adaptive Filters, New* York, New York, John Wiley & Sons, 1987.

[Tsividis *et al.,* 1986]
Y. Tsividis, M. Banu, and J. Khoury, "Continuous-Time MOSFET-C Filters in VLSI", *IEEE Trans. on Circuits and Systems,* vol. CAS-33, pp. 125-140, Feb. 1986.

[Viswanathan, 1986]
T.L. Viswanathan, "CMOS Transconductance Element", *Proceedings of the IEEE,* vol. 74, pp. 222-224, Jan. 1986.

[White, 1975]
S.A. White, "An Adaptive Recursive Digital Filter", *Proc. 9'th Asilomar Conf. Circuits Syst. Comput.,* pp. 21-25, Nov. 1975.

[Widrow and Hoff, 1960]
B. Widrow and M.E. Hoff, "Adaptive Switching Circuits", *IRE WESCON Conv. Rec.,* pt. 4, pp. 96-104, 1960.

[Widrow and McCool, 1977]
B. Widrow and J.M. McCool, "Comments on 'An Adaptive Recursive LMS Filter"', *Proc IEEE,* vol. *65,* pp. 1402-1404, Sept. 1977.

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

[Widrow et al., 1967]
B. Widrow, P.E. Mantey, L.J. Griffiths and B.B. Goode, "Adaptive Antenna Systems", *Proceedings of the IEEE,* vol. *55,* pp. 2143-2159, Dec. 1967.

[Wonham, 1985]
W.M. Wonham, *"Linear Mu&variable Control",* 3rd. Ed., New York, Springer-Verlag, 1985.

[Yassa, 1987]
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.

[Ziemer and Tranter, 1976]
R.E. Ziemer and W.H. Tranter, ***Systems, Modulation, and Noise,*** Boston, Mass., Houghton Mifflin Company, 1976.