DIGITAL FILTERS

AND

CASCADE CONTROL COMPENSATORS

Alan Graham Bolton

Department

of

Electrical and Electronic Engineering

University of Adelaide

May. 1990

TABLE OF CONTENTS

ABSTRACT

The thesis contains new design procedures for digital filters and cascade control compensators. Fixed point binary representation of signals is considered, as is the sequential execution of a programmed device. The two's complement fixed point binary arithmetic has specific nonlinearities. Sequential program execution gives intrinsic timing properties which make certain filter structures relatively convenient to implement.

A versatile second order digital filter structure is proposed, a feature being that the notch depth is infinite at all sampling rates. New and convenient design procedures to compute the required coefficient values are developed. One procedure gives ideal magnitude equivalence with frequency warping, whilst another gives approximate magnitude equivalence without frequency warping. The thesis also contains a derivation from first principles for Bessel digital filters.

Then the effects of representing the signal using two's complement arithmetic are considered. Techniques are devised to avoid both small and large signal limit cycles. These techniques apply to filters where the input is not zero.

The thesis also contains new design techniques for sampled control systems which allow design evaluations to be made within the Laplace domain. One aspect is to extend compensation for frequency warping of the bilinear transform from the frequency axis to any point in the Laplace domain. Also a method for sequencing instructions so as to minimize computational delay is presented.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University and that, to the best of the candidate's knowledge and belief, the thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis,

and,

the author consents to the thesis being made available for photocopying and loan.

Alan Bolton

ACKNOWLEDGEMENTS

The author is grateful for the considerable assistance given by his supervisors, Dr. B.R. Davis and Professor R.E. Bogner. In one area the contribution was so significant that Dr. Davis shared the authorship of the paper. Guidance was freely given during the entire candidature, and it was invaluable.

Dr. Tran Thong of the Digital Research Laboratories at Tektronix, Beaverton U.S.A, made helpful suggestions, particularly in the areas of filter simulation and transfer function availability. The extensive computing and administrative support of Tektronix' laboratories are acknowledged also.

The South Australian Institute of Technology has supported the investigations. The facilities to study dynamic systems, developed by Mr. J. Mount, Senior Lecturer, were very useful. The author also wishes to acknowledge the support of the Head of the School of Electrical Engineering, Professor E. Hobson, and Professor E.W. Mills who, as Director of the South Australian Institute of Technology, approved the Ph.D. candidature.

Professor B.Albano and Professor B.W.Williams assisted considerably by proof reading the thesis.

LIST OF SYMBOLS

a	coefficient in z domain transfer function, and in the forward path of the digital filter structure.
a_B	constant of the bilinear transform.
a_C	constant within the transfer function of the sampled compensator.
$-a_s$	real component of location of singularities in the Laplace domain.
$-a_y$	real component of location of singularities in the y domain.
$a_0, a_1, a_2 \ldots$	coefficients of the all pole analog digital filter.
a^*	one of the coefficients in the forward path of the proposed digital filter structure.
a'	coefficient of a quadratic in s'.
A	matrix used in the Bessel filter design such that its elements a_i are the coefficients of the canonic filter.
angle[a(ø)]	Angle of a complex function, as a function of frequency
b	coefficient in z domain transfer function, and in the damping loop of the digital filter structure.

b_B constant of the bilinear transform.

b_C constant within the transfer function of the sampled compensator.

$\pm jb_s$ imaginary component of location of Laplace domain singularities.

$\pm jb_y$ imaginary component of location of y domain singularities.

b' coefficient of a quadratic in s'.

$B(z)$ the bilinear transform.

$c(nT)$ time domain output of the filter.

$C_s(s)$ continuous output function of the control system which has no samplers.

$C_s^*(s)$ sampled output function of the sampled control system, Laplace domain.

$C_z(z)$ sampled output function of the sampled control system, z domain.

$cis(\phi nT)$ $= \cos(\phi nT) + j\sin(\phi nT)$

e_i input function to the cascade compensator which is also the error function of the control system.

$F[\exp(s_d T)]$ a general function of the design point, s_d.

$g_1, g_2, g_3,$ constants used to describe a class of functions examined for their suitability as a Liapunov function.

G	matrix used in the Bessel filter design so that for an Nth order filter its elements are $g_{ij} = (i+k)^{2N-1}$, where k is the desired delay of the Bessel filter, (equation 2.5.15).
$G_s(s)$	transfer function of the plant of the control system, continuous (without sampling).
$G_s^*(s)$	transfer function of sampled plant, Laplace domain.
$G_s^*(j\phi)$	frequency domain function of $G_s^*(s)$.
$G_z(z)$	sampled digital plant, z domain.
$G_s'(s)$	transfer function of continuous compensator.
$G_s'^*(s)$	transfer function of sampled compensator, Laplace domain.
$G_z'(z)$	sampled digital compensator, z domain.
H	matrix used in the Bessel filter design with elements $h_i = k^{2i-1}$, where k is the desired delay of the Bessel filter, (equation 2.5.16).
$H_s(s)$	Laplace domain transfer function used to derive magnitude invariant design procedure.
$H_s(j\phi)$	frequency response of $H_s(s)$.
$[H_s(j\phi)]^*$	sampled frequency response of $H_s(j\phi)$.
$H'_s(s'T)$	transfer function of introduced Laplace operator s'.
i	summation parameter.
k	ratio of ϕ_e to ϕ_c.

k_B constant of the bilinear transform.

k_C constant within the transfer function of the sampled cascade compensator.

kT desired time delay of the Bessel response.

Lf(X) Liapunov function of the state variables X.

$Lf(x_1,x_2)$ Liapunov function of the state variables x_1 and x_2.

M magnitude term of transfer function $T_z(z)$, such that the highest powers of z are normalized.

±M bounds on the range of the signal.

M(ø) magnitude as a function of frequency of the filter response.

n an index variable in the summations to N.

N order of polynomial or system, or number of terms evaluated in the frequency domain summation transfer function of the sample system.

p coefficient in the numerator of a z domain transfer function.

P number of poles of the control system plant transfer function, $G_s(s)$.

q coefficient in the numerator of a z domain transfer function.

quant(x) the two's complement quantization characteristic.

q^{-1} unit time domain delay operator.

Q	quality factor of the complex pole pair, ($Q = 1/2\beta$).
$r(nT)$	time domain input to the digital filter, or to the control system.
$R_s(s)$	continuous input function to the control system which has no samplers.
$R_s^*(s)$	sampled input to the control system, Laplace domain.
$R_z(z)$	sampled input to the control system, z domain.
s	the sign of the overflow in a signal variable, $s = 1$ for a positive overflow and $s = -1$ for a negative overflow.
s_d	Laplace domain design point of control system, (location of dominant closed loop poles, impulse invariant with z_d).
s'	Laplace domain operator corresponding to the frequency variable ϕ'.
T	sampling interval. When normalized $T = 1$.
$T_\beta(\beta)$	transfer function of the filter as a function of β.
$T_s(j\phi)$	transfer function of $T_s(s)$ at frequency ϕ.
$T_s(s)$	transfer function (output/input) of a continuous time system.
$T_{s,AN'}(s)$	transfer function (Laplace domain) of an asymmetric notch.
$T_{s,B}(s)$	transfer function (Laplace domain) for band pass filter.

$T_{s,H}(s)$ transfer function (Laplace domain) for high pass filter.

$T_{s,L}(s)$ transfer function (Laplace domain) of low pass filter filter.

$T_{s,N}(s)$ transfer function (Laplace domain) of a symmetric notch, (which has equal steady state and asymptotic high frequency gains).

$T_s^*(s)$ sampled transfer function (output/input) of the sampled control system.

$T_y(y)$ transfer function of the filter as a function of y.

$T_z(z)$ transfer function of the filter structure, or transfer function (output/input) of the sampled control system.

$T_{1/z}(1/z)$ transfer function of the filter as a function of 1/z.

$T_{z,AN}(z)$ transfer function (z domain) of an asymmetric notch of infinite depth.

$T_{z,AN}'(z)$ transfer function (z domain) of an asymmetric notch of finite depth.

$T_{z,B}(z)$ transfer function (z domain) for band pass filter.

$T_{z,H}(z)$ transfer function (z domain) for high pass filter.

$T_{z,L}(z)$ transfer function (z domain) of a low pass filter.

$T_{z,ss}(z)$ steady state gain of $T_z(z)$.

v_B time domain signal in filter with the band pass response.

v_L time domain signal in filter with the low pass response.

x_{ovf} $x_{overflow}$, variable computed when arithmetic overflow has occurred.

x_{lin} x_{linear}, variable which would be computed if the arithmetic were linear.

x_{bnd} $x_{bounded}$, value present in the filter which has been designed so that that each variable in the filter is operating within its linear range.

x_{trn} $x_{transient}$, an additional component present in the computer which is not anticipated by the design. In general this transient is such that the bounds of linear operation are exceeded.

x_{sat} $x_{saturated}$, saturated value of the signal.

x_{dif} $x_{difference}$, difference between the saturated value and x_{lin} which would be computed by the linear system.

$x_{0,i}$ variable used in the proposed second order digital section which is computed during the i^{th} loop. This is the symmetrical notch output, (equation 3.3.87).

$x_{1,i}$ variable used in the proposed second order digital section which is computed during the i^{th} loop. This is the band pass output, (equation 3.3.89).

$x_{2,i}$ variable used in the proposed second order digital section which is computed during the i^{th} loop. This is the low pass output, (equation 3.3.90).

$x_{3,i}$ variable used in the proposed second order digital section which is computed during the i^{th} loop. This is the high pass output, (equation 3.3.88).

y operator for sampled domain, [Bolton 80a].

z_d z domain design point of control system, (location of dominant closed loop poles, impulse invariant with s_d, $z_d = \exp[s_d T]$).

Z number of zeros of the control system plant transfer function, $G_s(s)$.

$*$ symbol denoting a sampled function.

β damping factor of a complex pole pair.

βT digital differentiation operator proposed by Karwoski.

$\delta G_{s,N}^*(s_d)$ difference between the sampled transfer function of the plant of the system, $G_s^*(s_d)$ and the summation to N frequency domain terms for $G_s^*(s_d)$.

δT the delay between the input sample to the cascade compensator (error function, e_i) and the corresponding output (forcing function f_i). T is the total loop time.

σ damping component of the Laplace domain operator s.

σ_d damping component of the design point s_d.

$-\Gamma$ (real) location of the zeros of the lead compensator $H'_s(s)$ introduces in the magnitude invariant frequency domain design procedure which is introduced to compensates for the effect of a zero order hold.

ϕ frequency component of the Laplace operator s.

ϕ_c cutoff frequency of the analog low pass filter used in the magnitude invariant frequency transform.

ϕ_d frequency component of the design point of the Laplace domain.

ϕ_e frequency of equality of continuous and sampled responses, obtained using the bilinear transform, or using the magnitude invariant design procedure.

ϕ' frequency variable in the sampled domain related to a frequency in the continuous system, ϕ, by a frequency warping characteristic.

σ location of real Laplace domain pole for the magnitude invariant frequency domain design procedure.

τ time constant of first order response, and inverse of the natural frequency of complex pole pair.

Ω_n natural frequency of a complex pole pair.

Ω_e frequency of equivalence between the analog and the derived digital filter obtained using the bilinear transform.

Ω_{3dB} frequency when the response is 3dB less than the asymptotic pass band response.

Chapter 1 PROPOSED DIGITAL FILTER STRUCTURE

1.1 Introduction

There are many applications where a microprocessor is used to process signals. The data which is processed by the filter may be stored in digital form, or it may be samples of a real time signal. The output of the digital filter may be sent to an analog to digital converter or stored digitally. There are requirements for routine design procedures for these filters. The objective of this thesis is to present a versatile filter structure and convenient design procedures. These designs require

1. a versatile filter structure, capable of implementing a wide variety of transfer functions. (chapter 1).
2. procedures to give the coefficients of these filters, (chapter 2), and,
3. program steps to avoid the limit cycles associated with arithmetic non-linearities, (chapter 3).

Further aspects of digital signal processing arise when the digital filter is part of a closed loop control system which includes a given analog plant. In these applications the digital filter is called a cascade compensator in a control system. Chapter 4 presents an extension to the design procedures for sampled cascade control compensators when their design is based on an analog approximation.

1.2 Existing Digital Filter Design Procedures.

The context of the proposed design procedures for digital filters can be illustrated using an example of a conventional design of the second order canonic structure, (figure 1.1). The transfer function of this structure is

$$T_z(z) = M \frac{1 + qz^{-1} + pz^{-2}}{1 - bz^{-1} + az^{-2}} \qquad \ldots\ 1.2.1$$

A typical requirement for a filter is a second order low pass Butterworth response which has a cutoff (3 dB) frequency of 20 Hz. The design procedures can be illustrated by considering a sampling frequency of 1 kHz. The bandwidth of this filter, 20 Hz, is small in relation to the sampling frequency. The poles of the digital filter are, however, heavily damped because these correspond to the poles of a second order Butterworth filter. Therefore this narrow band filter has poles with a low Q.

The transfer function of the analog second order Butterworth filter is

$$T_s(s) = \frac{\Omega_n^{\ 2}}{s^2 + 2\beta\Omega_n s + \Omega_n^{\ 2}} \qquad \ldots\ 1.2.2$$

where $\Omega_n = 2\Pi.20$ rad/s and $\beta = 1/\surd 2$.

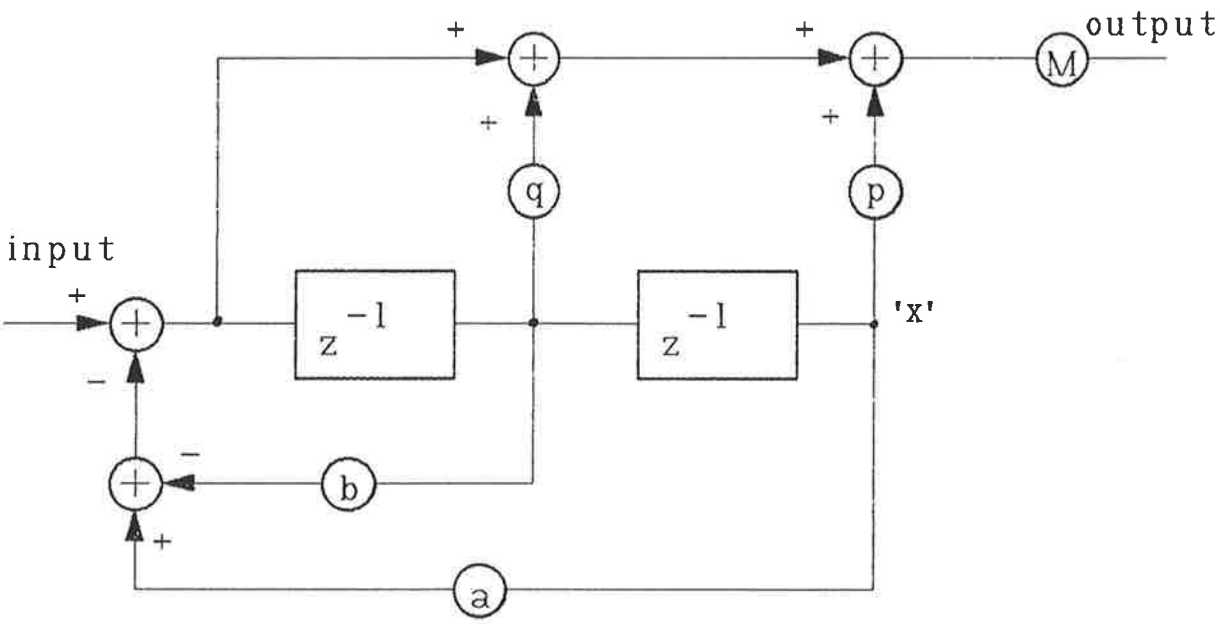


Figure 1.1 **Canonic z Domain Second Order Digital Filter.**

The bilinear transform is

$$sT = k_B \frac{z - 1}{z + 1} \qquad \ldots\ 1.2.3$$

where

$$k_B = \Omega_e T \cot(\Omega_e T/2) \qquad \ldots\ 1.2.4$$

Ω_e being the frequency of equivalence between the analog and the derived digital filter. In this design procedure this frequency is chosen to be the edge of the pass band. For this Butterworth filter Ω_e is Ω_{3dB} which in turn is Ω_n.

The numeric results for this example are

$$k_B = 1.997,\ a = 0.8372,\ b = 1.823,$$

$$p = 1.000,\ q = 2.000 \text{ and } M = 0.003622 \qquad \ldots\ 1.2.5$$

Two aspects of this design can be improved for many implementations. The first relates to the signal magnitudes, and the second to reducing the number of multiplications below five.

1.2.1 Signal Magnitude

When programming this structure in a microprocessor which has fixed point arithmetic it is important to consider the magnitudes of the signals. The final value theorem can be used to find the steady state magnitudes. The final value theorem states that the steady state response of a system $T_z(z)$ to a unit step is the limit of $T_z(z)$ as z approaches 1. In this example, if an input of 1 unit is applied to the filter the magnitude of the signal at point 'X' is

$$\underset{z \text{ approaches } 1}{\text{Lt}} T_z(z) = \frac{1}{1 - bz^{-1} + az^{-2}} \qquad \ldots 1.2.6$$

$$= 69.02$$

Hence, in the steady state, the signals through the delay path are more than 2^5 times larger than the input signal. With limited register lengths this difference requires careful magnitude scaling of the signals. The scale factor depends on the location of the z domain poles which are to be implemented. To compensate for this magnitude difference this filter requires a magnitude scale factor, M, of 0.003622 which is less than 2^{-8}.

1.2.2 Feedforward Terms

The multiplications by coefficients p and q give the zeros of transmission. The polynomial of the numerator is

$$z^2 + 2z + 1 \quad \ldots 1.2.7$$

and hence two zeros of transmission are at $z = -1$. These zeros arise when the bilinear transform is applied to an all pole analog transfer function.

In practice filters such as the chosen example have sampling rates which are high in relation to the cutoff frequency. In consequence the locations of the z domain poles are near $z = +1$. To achieve these pole locations a large proportion of the signals are fed back around the delay loop. One consequence is that the signal magnitudes through the delay loop are large in relation to the input signal. A second consequence is that the signals through the delay elements (at 'X') do not change rapidly with signals of interest. The z domain representation of this fact is that the zeros at $z = -1$ are removed from the region near $z = +1$ which contains both the poles of the filter and its pass band. In consequence the feedforward coefficients p and q are operating on signals which have very similar magnitudes. If p and q were set to zero, the zeros of transmission at $z = -1$ would not be implemented, but in many applications there would not be a significant change in the transfer function. Yet there would be a saving of two multiplications. (It should be noted that coefficients p and q are convenient to implement, being factors of 1 and of 2). There is scope to develop designs for filters which do not require these zeros of transmission. These filters are all pole in the z^{-1} domain.

1.2.3 Limit Cycles

Once the recursive filter has been built using two's complement binary arithmetic it is possible that both large and small signal limit cycles will arise. It is necessary to include program steps which remove these effects.

1.3 Contributions of the Thesis, (Filters)

The elementary example of the canonic implementation of a Butterworth filter which has been obtained using the bilinear transform can be used to illustrate the purpose of the contributions of this thesis.

1.3.1 Filter Structure, (Chapter 1)

A filter structure [Bolton '81c] was obtained using equivalence with analog computer implementations of a versatile second order resonant section. The second order section to be presented has unity steady state gain because the feedback gain is unity and a pure integration is present in the forward path. The timing of the derived digital filter section is suited to the sequential implementation of a program. These techniques have already been applied to digital filters, [Kingsbury '72], but are extended here to give a high pass and notch outputs, and variations suited to high Q and to low Q filter sections.

1.3.2 Design of the Filter Coefficients, (Chapter 2)

The example of section 1.2 illustrates that applications areas exist where feedforward terms give little enhancement to the filter's performance because the resulting zeros of transmission are considerably removed from the poles of the transfer function and its pass band region. In these situations there is a requirement to provide design procedures where the derived transfer functions are all pole in the z^{-1} domain. Three new design procedures are devised for these situations. These involve a consideration of the magnitude of the transfer function [Bolton '84a], and also the derivation from first principles of a Bessel (maximally flat group delay) digital filter, [Bolton '84d].

A new sampled operator, y, [Bolton '80a], is proposed to help examine sampled systems. The y operator approaches the Laplace operator 's' at high sampling rates where sT approaches zero.

1.3.3 Limit Cycles, (Chapter 3)

Liapunov analysis is applied to the filter to derive program steps which eliminate both large and small limit cycles which are associated with arithmetic non-linearities. A feature of the analysis is that transients in nonlinear systems are considered.

1.4 The Proposed Structure

The majority of microprocessors implement two's complement arithmetic most conveniently. This arithmetic is considered here. It is necessary to avoid signal overflow and yet maintain the signals levels near maximum levels so as to minimize the effects of signal quantization. Therefore the filter structure must have signal levels within the filter near maximum levels for a wide range of singularity locations.

The digital filter structure [Bolton '81c] is based on an analog computer equivalent. The signals in analog computers are voltages of limited range, usually between +10 and -10 V. Any signal overflow causes the analog simulation to be in error. At the same time offsets and noise are present at the inputs of the amplifiers, so that operating at small signal levels introduces unacceptable errors. The analog computer, like the microprocessor with fixed point arithmetic, must maintain all signal levels near but not exceeding the maximum values. Analog computing techniques have evolved to suit the situation where signals have a limited dynamic range. It is quite straight-forward to derive corresponding digital implementations by substituting digital Euler integration for analog integration.

Figure 1.2.A illustrates a conventional analog computer technique for obtaining a complex pole pair. Figure 1.2.B shows an equivalent circuit diagram which is the conventional biquad filter. The maximum signal levels throughout these implementations are at nearly the same values for a wide range of singularity locations. The features which provide this property are:

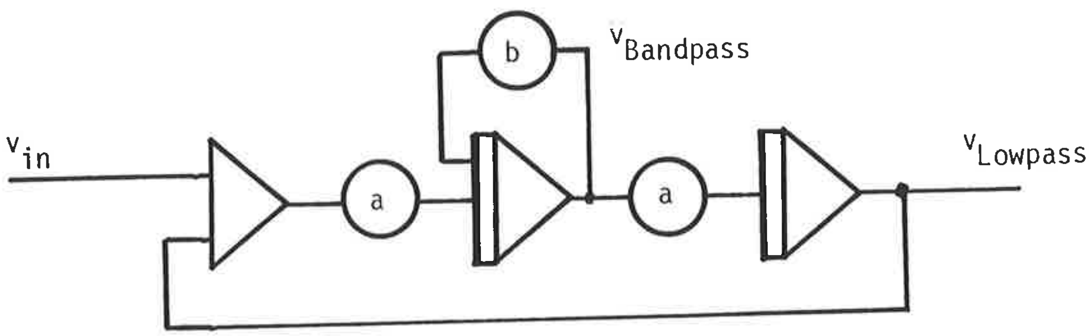
i. the gains around the loop are distributed uniformly. In particular the gains of the integrators are equal,

ii. there is a pure integration in the forward path, and

iii. the system has a gain in the feedback path of unity.

The use of unity gain in the feedback path with a pure integration in the forward path ensures that the dc gain of the section is unity. This can be expressed in the notation of control theory. The type number of the system is one because there is one pure integration in the loop, so, with unity feedback, the step response has zero steady state error.

A.



B.

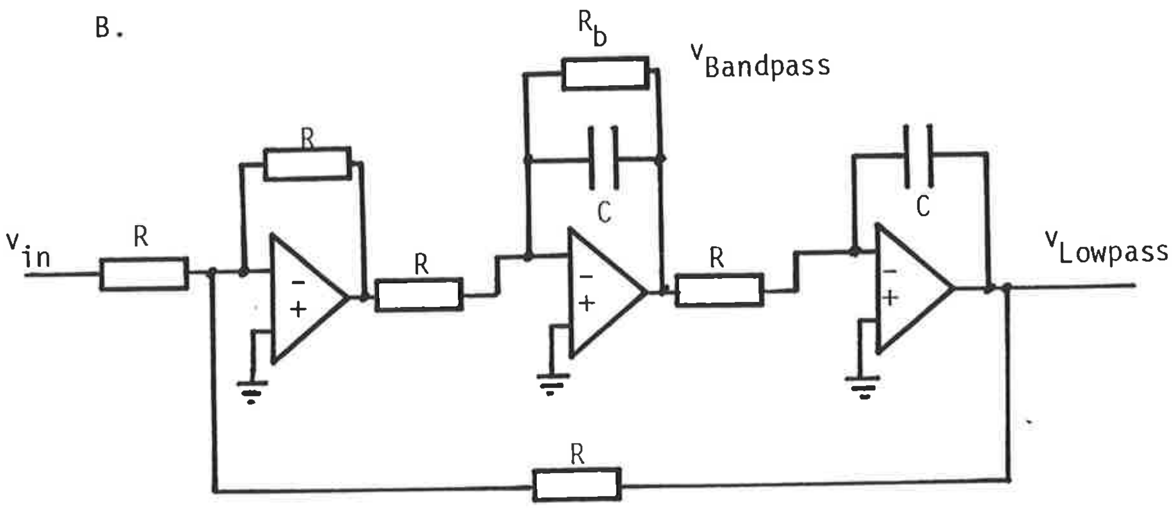


Figure 1.2 Analog Computer Implementation of a Complex Pole Pair with its Biquad Equivalent.

Although the steady state gain of this filter is unity, at very low damping factors the magnitude of the signals within the filter will exceed that of the input signal by a factor approximating the Q of the resonator. It is possible to alter the structure to accommodate this situation. This is one of the variations on the structure described later in this chapter.

The digital filter can be obtained by replacing the analog integration of the analog computer by the simplest of digital integrations, Euler. Such a filter has been devised, [Kingsbury '72]. An aspect of the contribution here is to provide both notch and high pass outputs. The analog computer implementation of figure 1.3 illustrates that the damping term can be implemented before the feedback loop. This particular sequence provides both high pass and symmetric notch responses as signal variables of the filter. Three representations of this filter are presented, the analog computer diagram (figure 1.3), the block diagram, (figure 1.4), and the program sequence (figure 1.5).

The relationship between the three representations is not trivial. The block diagram is a convenient representation of a z domain expression. The z domain and hence block diagram representations presuppose that the signal variables are present only at the sampling instants, and that the operations on these variables occur instantaneously at these instants.

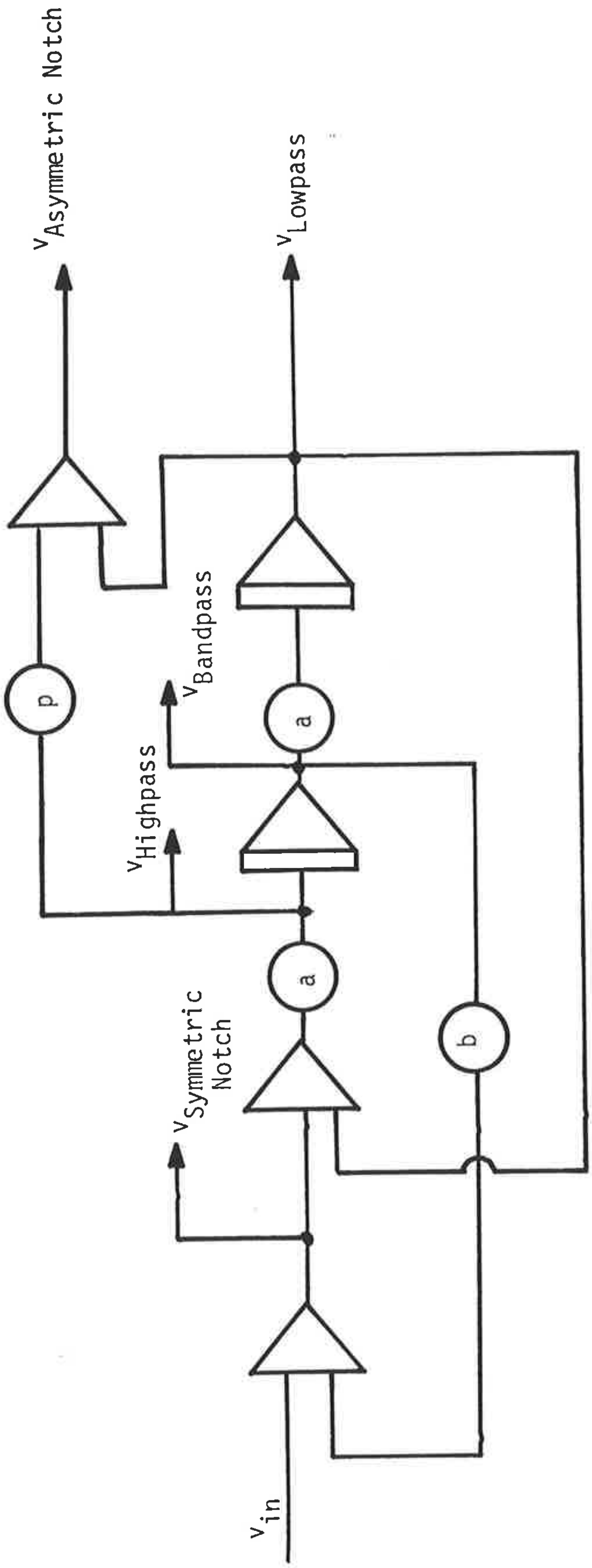


Figure 1.3 Alternative Analog Computer Implementation which gives Notch and High Pass Outputs.

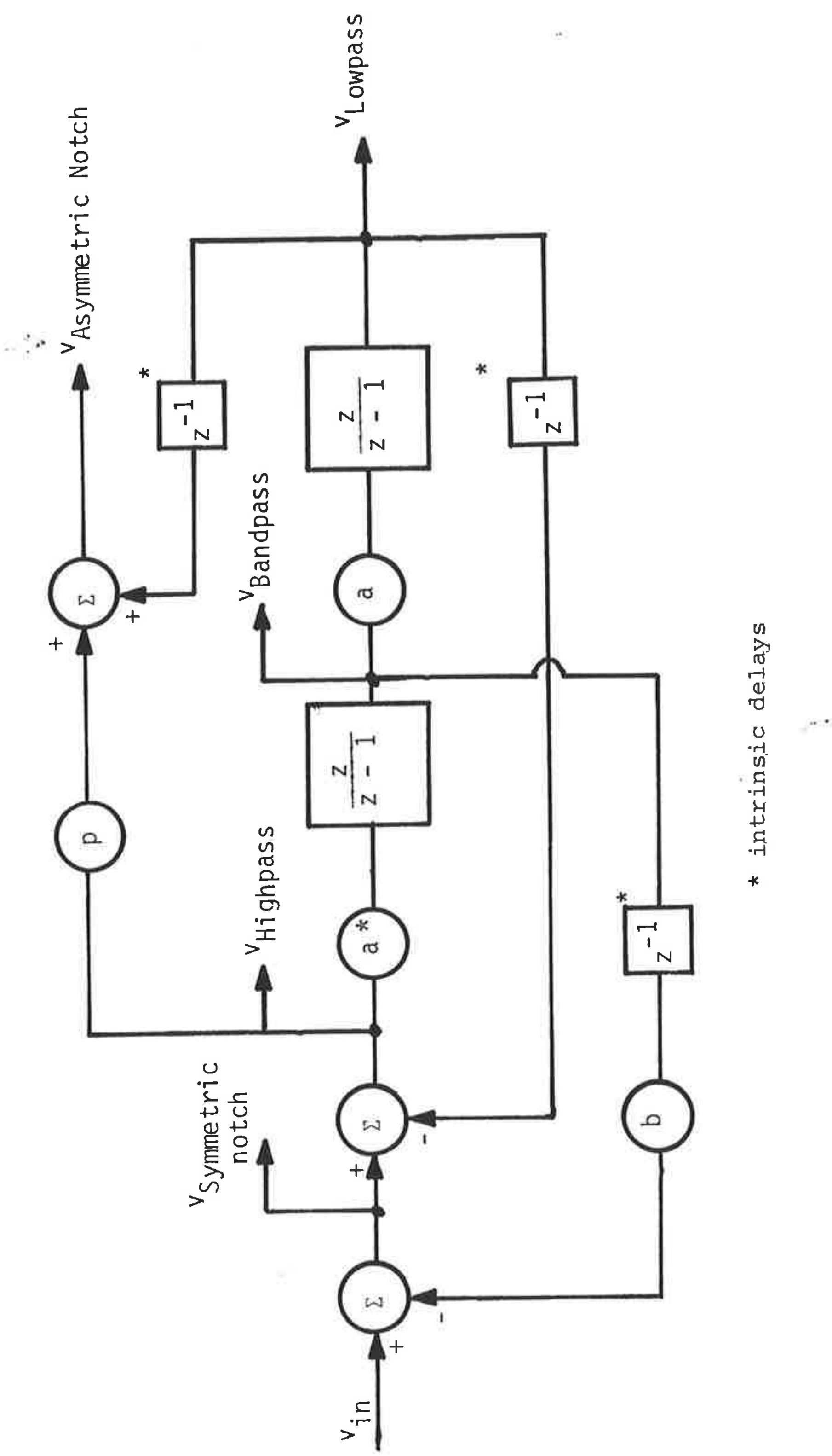
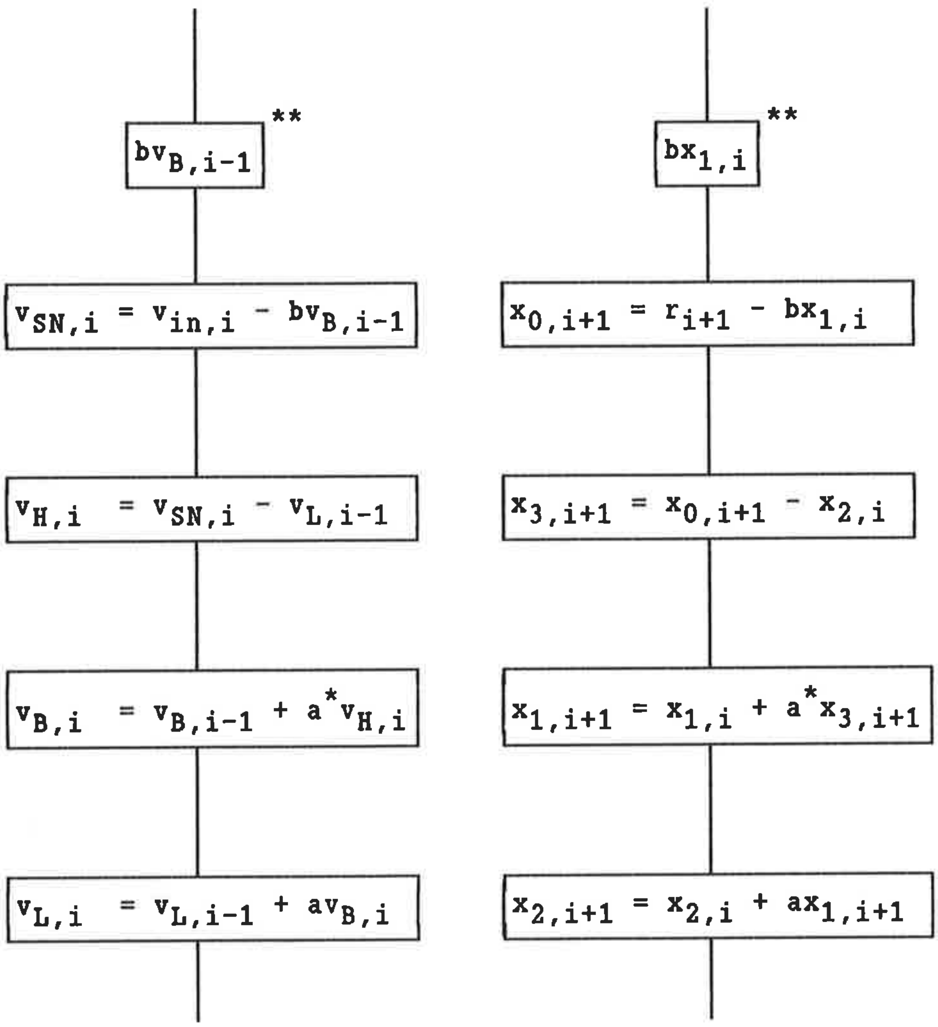


Figure 1.4 Block Diagram of the Proposed Filter Section.

[The coefficient a^* may differ from coefficient a]



** multiplications for the damping loop

Figure 1.5 Program Flowchart for the Proposed Filter Section.

In fact programs are executed sequentially. The variables in the computer are evaluated at various times during the sampling interval. There is an implied mapping of the actual computation times onto the simultaneous and instantaneous representation of the z domain. This mapping involves shifts in time.

One consequence of the differing representations is that it is readily possible to represent a delay free loop in block diagram form though there is no program equivalent. There is no means of actually implementing a delay free loop, although it can be represented on a block diagram. Similarly, whenever a variable is accessed during a program, and that variable was computed during the previous loop, there is an intrinsic delay in the feedback path. This delay does not require special program steps but does appear on the block diagram. The intrinsic delays for the proposed filter structure are given in the block diagram of figure 1.4.

1.4.1 Low Pass Response

The transfer function of the low pass filter of figures 1.3 to 1.5 is

$$T_{z,L}(z) = \frac{a^2 z^2}{z^2 - z\{2-a(a+b)\} + (1-ab)} \qquad \ldots\ 1.4.1$$

Using the final value theorem, the steady state gain of the transfer function is found by letting $z = 1$. This gain is unity.

When the damping coefficient $b = 0$ the magnitude of complex z domain roots is unity, illustrating that marginal stability is obtained when the damping coefficient $b = 0$. This is quite a significant result. The original analog filter is marginally stable when the damping coefficient is zero. This particular derived digital structure has the same property for all sampling rates of engineering interest, $(0<a<2)$. The property arises because the effects of digital integration have been compensated by the single intrinsic delay in the feedback path. One consequence of this similarity is that the response of the derived digital filter section closely approximates that of the original analog at high Qs. This aspect is discussed in the section giving the design procedures.

1.4.2 Band Pass Response

For continuous systems the band pass response, $T_{s,B}(s)$ is obtained by differentiating the low pass response, $T_{s,L}(s)$. For the continuous systems where the differentiation operator is 's', the relationship is

$$T_{s,B}(s) = sT_{s,L}(s) \qquad \text{... 1.4.2}$$

Similarly the band pass output for a digital filter is obtained by digitally differentiating the low pass output. This gives

$$T_{z,B}(z) = [1 - z^{-1}]T_{z,L}(z) \qquad \ldots\ 1.4.3$$

It is convenient to incorporate a scale factor of '1/a' so that the signal magnitude of the band pass signal is comparable with that of the low pass signal. In this case

$$T_{z,B}(z) = \frac{az(z-1)}{z^2 - z\{2-a(a+b)\} + (1-ab)} \qquad \ldots\ 1.4.4$$

This is the transfer function of the proposed digital filter section at the band pass output.

1.4.3 High Pass Response

The high pass response of the continuous system, $T_{s,H}(s)$, is related to the band pass response, $T_{s,B}(s)$ by the differentiation operator s, such that

$$T_{s,H}(s) = sT_{s,B}(s) \qquad \ldots\ 1.4.5$$

The biquad filter chosen here has two cascaded integrations so that high pass output signal is available as an internal signal. It is present at the input to the first integrator. The relationship to give the high pass output for a digital filter using digital differentiation is

$$T_{z,H}(z) = [1 - z^{-1}]T_{z,B}(z) \qquad \text{... 1.4.6}$$

Again it is convenient to incorporate a scale factor of '1/a' so that the signal magnitude of the band pass signal is comparable with that of the low pass. The high pass response is

$$T_{z,H}(z) = \frac{(z-1)^2}{z^2 - z\{2-a(a+b)\} + (1-ab)} \qquad \text{... 1.4.7}$$

1.4.4 Symmetric Notch

The transfer function of a notch filter has zeros of transmission at a finite frequency. With analog filters the zeros are on the frequency axis, whereas with sampled systems the corresponding zeros of transmission are on the unit circle in the z domain.

The term symmetric notch is used here to indicate that the responses at arbitrarily low frequencies and at arbitrarily high frequency are equal in magnitude.

The transfer function of the filter at the symmetric notch output is equal to the sum of the high pass to the previous low pass transfer functions.

$$T_{z,N}(z) = T_{z,H}(z) + z^{-1}T_{z,L}(z) \qquad \ldots 1.4.8$$

$$= \frac{(z-1)^2}{z^2 - z\{2-a(a+b)\} + (1-ab)} + \frac{a^2z^1}{z^2 - z\{2-a(a+b)\} + (1-ab)} \qquad \ldots 1.4.9$$

$$= \frac{z^2 - (2-a^2)z + 1}{z^2 - z\{2-a(a+b)\} + (1-ab)} \qquad \ldots 1.4.10$$

The quadratic of the numerator of equation 1.4.10 gives the locations of the zeros of transmission of the transfer function. The magnitudes of these roots are unity, indicating that the finite zeros of transmission are on the unit circle. The roots have a real component of $(1 - a^2/2)$ so they occur at

$$z = 1 \pm \text{angle}\{\arccos[1-a^2/2]\} \qquad \ldots 1.4.11$$

1.4.5 Asymmetric Notch

The importance of the sequence of program steps of the filter's transfer function can be illustrated using the asymmetric notch. This output is required for the implementation of elliptic analog filters which have been transformed for sampled implementation using the bilinear transform.

With analog filters a notch response is obtained as a weighted sum of the low pass and the high pass responses. In this case

$$T_{s,AN'}(s) = pT_{s,HP}(s) + T_{s,LP}(s) \quad \ldots 1.4.12$$

$$= \frac{ps^2}{s^2 + 2\beta\Omega_n s + \Omega_n^2} + \frac{\Omega_n^2}{s^2 + 2\beta\Omega_n s + \Omega_n^2} \quad \ldots 1.4.13$$

$$= \frac{ps^2 + \Omega_n^2}{s^2 + 2\beta\Omega_n s + \Omega_n^2} \quad \ldots 1.4.14$$

The roots of the numerator polynomial have no term in s^1, so they are located on the frequency axis, at $s = \pm j\Omega_n/\surd p$.

With the digital filter there are two ways of combining the low pass with the high pass responses. It is possible to perform the addition at the completion of the computation loop.

In this case the notch output is

$$T_{z,AN'}(z) = pT_{z,H}(z) + T_{z,L}(z) \quad \ldots 1.4.15$$

$$= \frac{p(z-1)^2}{z^2 - z\{2-a(a+b)\} + (1-ab)} + \frac{a^2z^2}{z^2 - z\{2-a(a+b)\} + (1-ab)} \quad \ldots 1.4.16$$

$$= \frac{p(z-1)^2 + a^2z^2}{z^2 - z\{2-a(a+b)\} + (1-ab)} \quad \ldots 1.4.17$$

The location of the zeros of transmission of the transfer function are given by the roots of the quadratic on the numerator of equation 1.4.17. The coefficient of z^0 is p whereas the coefficient of z^2 is $(p+a^2)$. Since these coefficients are not equal the roots of the quadratic do not have unit magnitude. Although the original analog filter has zeros exactly on the imaginary axis in the Laplace domain, these zeros of transmission are not on the unit circle of the z domain. In order to shift the z domain zeros of transmission so that they are located on the unit circle it is possible to combine an additional term from the band pass output. This requires an additional multiplication, with its associated execution time and quantization noise.

It is possible to choose the sequence of program steps so that when the high pass and low pass outputs are combined in such a way that the zeros of transmission are on the unit circle. This is a property of the zeros if the high pass output is added to the low pass output of the previous loop. This addition does not require any additional program steps. The program sequence of figure 1.6 is used. An intrinsic delay is present in the low pass output since it is recalled from the previous computation loop.

In this case

$$T_{z,AN}(z) = pT_{z,H}(z) + z^{-1}T_{z,L}(z) \qquad \text{... 1.4.18}$$

$$= \frac{p(z-1)^2}{z^2 - z\{2-a(a+b)\} + (1-ab)} + \frac{a^2z^1}{z^2 - z\{2-a(a+b)\} + (1-ab)}$$

... 1.4.19

$$= \frac{p(z-1)^2 + a^2z^1}{z^2 - z\{2-a(a+b)\} + (1-ab)} \qquad \text{... 1.4.20}$$

Again the location of the zeros of transmission of the transfer function are given by the roots of the quadratic on the numerator of equation 1.4.20. In this case the coefficient of z^2 is p as is the coefficient of z^0. Hence the roots lie on the unit circle in the z domain. It follows that this program sequence can be used to avoid the implementation of an additional multiplication from the band pass output.

The program sequence which gives the zeros of transmission on the unit circle is convenient to program on microprocessors with few registers. This is because the multiplication of the damping loop to give $bv_{B,i-1}$ is completed before there is the requirement to store the input signal, $v_{in,i}$, as illustrated in the program sequence of figure 1.5.

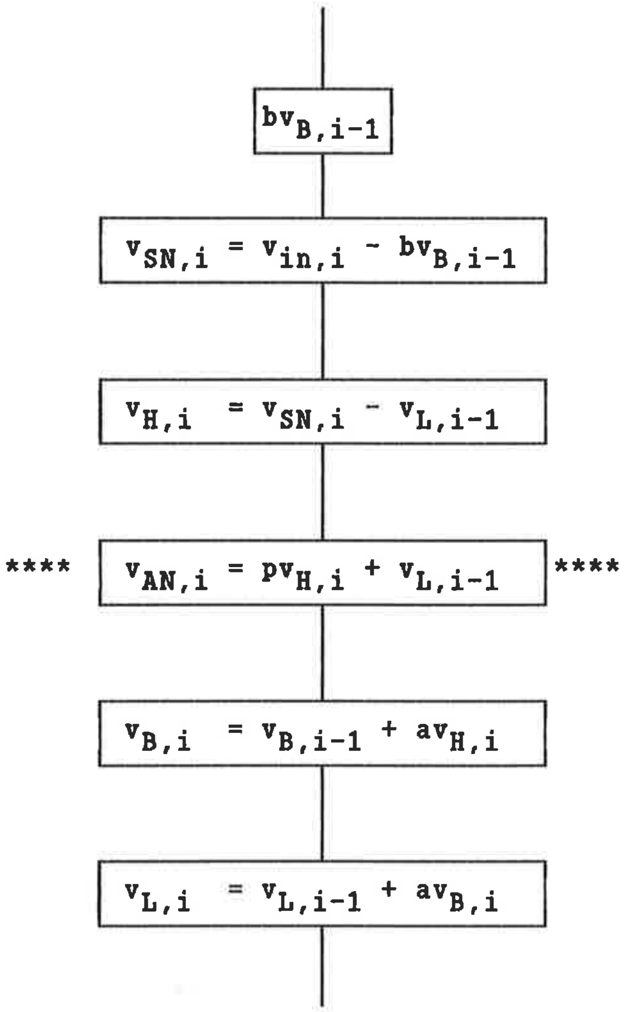


Figure 1.6 Filter Program with the Step Used to Obtain the Asymmetric Notch.

1.5 Context of the Structure

The digital filters considered here are those which use cascaded second order sections to obtain higher order transfer functions. Alternative approaches include wave and lattice structures. These filters suit many applications including those requiring dynamic changes to the filter's transfer function, [Fettweis and Meerkotter '75], and they exhibit excellent coefficient sensitivities, [Wegener '78]. However cascaded second order sections are considered here.

1.5.1 Canonic Structure

The structure which has been examined is a digital incremental filter, since the addition operator is used to compute its signal variables. This contrasts with the canonic digital filter which uses delay elements. It has been shown that the steady state gains of the signals of the canonic filter depend on the location of the singularities. These gains can be evaluated using the final value theorem in the z domain. (The steady state response to a step of unit magnitude is given by the value of the transfer function at z = 1). The transfer function for the second order canonic filter without magnitude scaling is

$$T(z) = \frac{1}{1 - az^{-1} + bz^{-2}} \qquad \text{... 1.5.1}$$

The steady state gain, found by setting z = 1, is

$$T_{z,ss}(z) = \frac{1}{1 - a + b} \qquad \ldots\ 1.5.2$$

Applications for digital filters involve sampling rates which are sufficiently large to avoid the effects of aliasing. In this case the z domain singularities are near z = +1 and the sum [1-a+b] approximates zero. The canonic structure has a high steady state gain which depends on the location of the poles.

In some microprocessor applications the canonic form does have advantages. The coefficients a and b multiply the same signal, albeit delayed, so there can be economies in the tests necessary to perform the multiplications. Implementations of canonic digital filters have been described, [Brafman et al. '78].

1.5.2 Digital Incremental Structures

Figure 1.7 gives three digital incremental filter structures. These structures are

i. Agarwal Burrus 2 structure, [Agarwal and Burrus '75],

ii. Abu-El-Haija, Shenoi and Peterson structure [Abu-El-Haija et al, '78].

iii. Kingsbury '72.

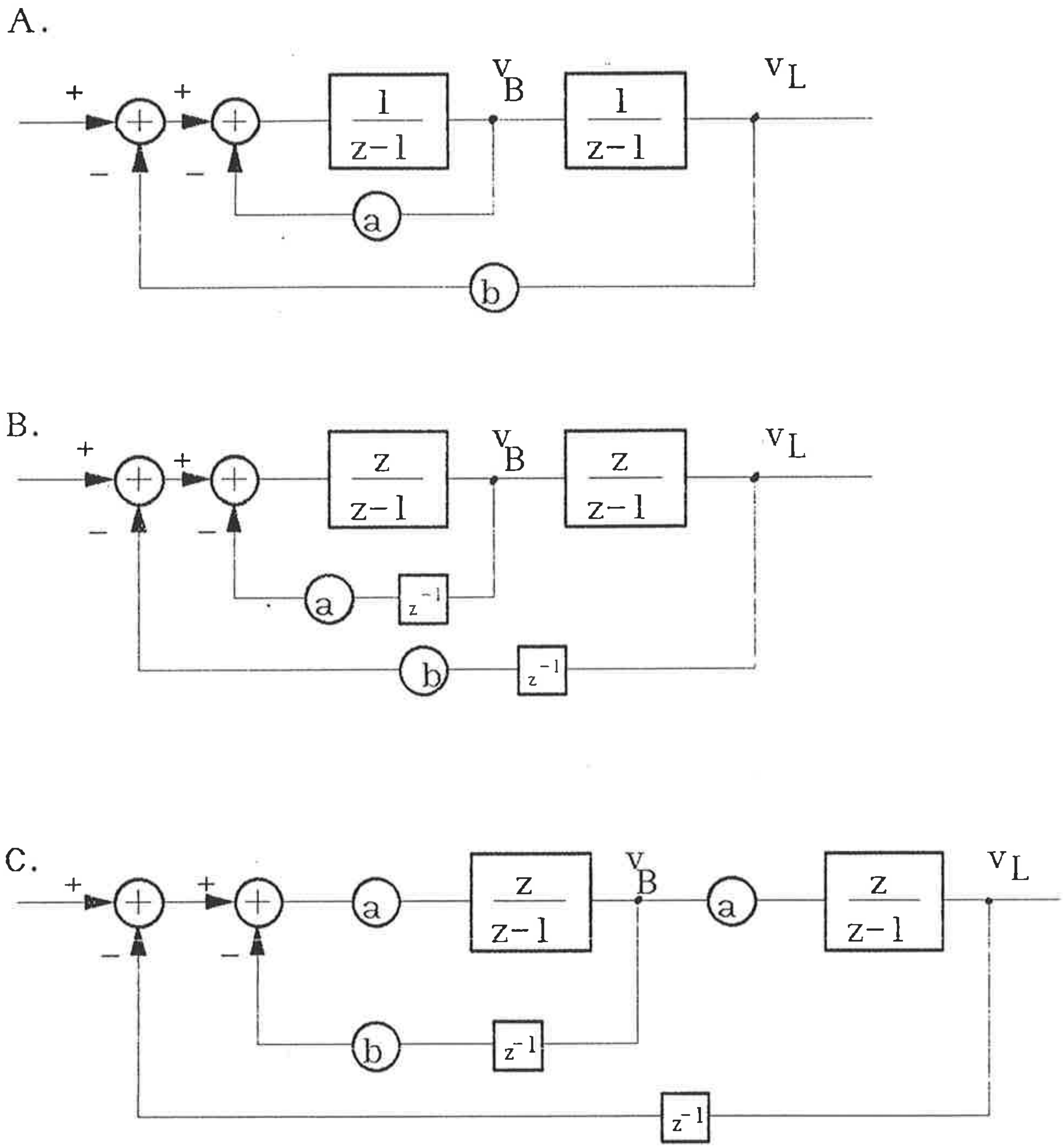


Figure 1.7 Comparison of Three Digital Incremental Filter Structures

i. Agarwal Burrus 2

ii. Abu-El-Haija, Shenoi and Peterson

iii. Kingsbury

The first two structures both use the coefficients in the feedback path. The result is that the steady state gains of the filters are much greater than unity. These structures exhibit considerable differences in the magnitudes of the signals, and scaling would normally be required in fixed point implementations.

These structures do differ because of their timing. The Agarwal Burrus 2 structure is suitable for pipeline implementation since each computation only uses results from the previous computation loop. By contrast the Abu-El-Haija, Shenoi and Peterson structure uses the updated value of v_B to compute v_L.

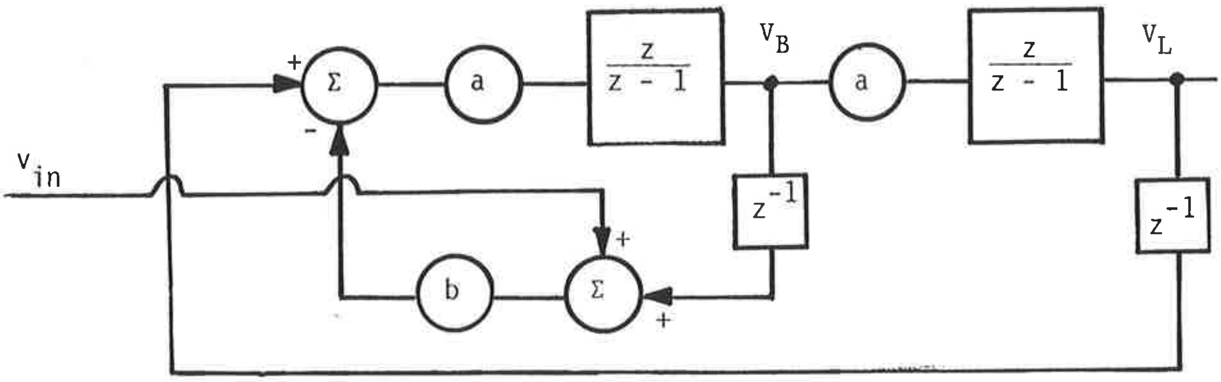
Whereas both the structures of figures 1.7.A and B have the coefficients in the feedback path, Kingsbury's structure of figure 1.7.C uses a pure integration in the forward path, distributes the forward path gain and uses unity feedback, [Kingsbury '72]. These aspects have been used in the structure presented here. The contributions have been to modify Kingsbury's structure so as to provide a high pass as well as notch responses. Alternative notch structures require an additional multiplication, [Kingsbury '72, Abu-El-Haija and Peterson '79].

1.6 Variations in the Proposed Structure

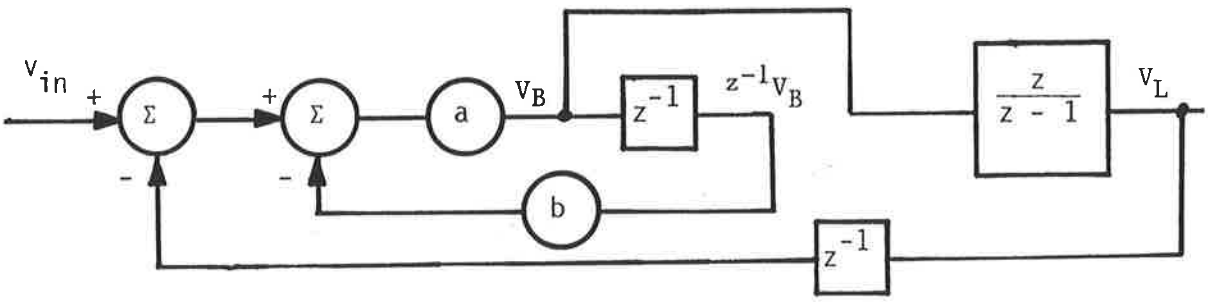
There can be many specialized applications when the general purpose filter section is not the most appropriate. These applications arise at both very high and very low damping factors. It is possible to adapt the structure to suit these special situations.

1.6.1 High Q

With high Q implementations the signal magnitudes within the filter are larger than those at the input. These signals are greater by a factor approximating the Q of the filter. In this case it is necessary to scale the input value by a factor approximating the damping coefficient b. Indeed it is possible to introduce the input variable before the damping coefficient b so as to provide this scaling without any additional multiplications. The filter structure is given in figure 1.8.A.



A.



B.

Figure 1.8 Variations on the Proposed Structure.

A. High Q

B. Low Q

1.6.2 Low Q

For low Q implementations with low sampling rates it is necessary to provide a relatively large damping term around the first integrator. To achieve this it is possible to omit the addition of the digital incremental variable which gives the band pass output. This structure (figure 1.8.B) uses a variable, v_B, which is delayed, whereas another variable v_L is computed using the digital incremental form. The filter is chosen so that there remains a pure integration in the forward path with unity feedback, so the steady state gain remains unity.

1.6.3 Unequal Integrator Gains

Whereas the Agarwal Burrus 2 and Abu-El-Haija structures use only two multiplications the proposed structure uses three. The additional multiplication is used to distribute the forward path gain of the filter. There can be advantages to allowing the coefficients at the integrator inputs to differ. A slight difference will not alter the maximum signal magnitudes significantly, and yet there will be a much greater number of transfer functions available given a fixed coefficient wordlength. Indeed at low sampling rates the coefficients labeled 'a^*' in figure 1.4 may be set to unity.

1.7 Summary

The proposed structure of figures 1.3 to 1.6 is an extension of the structure of Kingsbury. It has been extended to provide

i. band pass, high pass and notch outputs as signal variables within the filter, and

ii. a notch response which is obtained using special timing so that the zeros of transmission are on the unit circle. Alternative notch structures require an additional multiplication, [Kingsbury '72, Abu-El-Haija and Peterson '79].

The proposed second order filter structure of figure 1.5 and its extensions for both high and low Q implementations can be used to implement a wide range of transfer functions and are convenient to program. It remains to devise techniques to determine their singularity locations so their coefficients can be determined (chapter 2). Also additional program steps are required to avoid large and small signal limit cycles, (chapter 3).

Chapter 2 DESIGN PROCEDURES

In the chapter 1 a structure was proposed which is suitable for implementation by a sequential program which uses fixed point binary representation of the signals. New design procedures which enable the engineer to determine the coefficients of the digital filter are presented in this chapter.

2.1 Aspects of Design Procedures For Digital Filters

2.1.1 Iterative Procedures

The objective when developing the new design techniques has been to provide procedures which are convenient to implement. Iterative techniques have been avoided because these are more difficult to implement than simple closed form algebraic expressions. Iterative procedures do suit demanding filter applications where the design is implemented by a specialist. These procedures are very flexible. They enable the simultaneous consideration of both time and frequency domains [Chottera and Jullien '82], compensation for zero order holds [Antoniou '82], for finite coefficient wordlengths [Acha and Robles-Diaz '81] and approximations to arbitrary magnitude responses [Thajchayapong and Rayner '74].

However the design procedures presented here are convenient for the non-specialist to implement. They can be expressed as simple equations, and are intended for routine situations where the specification of frequency discrimination waveform distortion does not require relatively complex iterative design techniques.

2.1.2 Frequencies Introduced by Sampling

When designing analog filters the frequency domain property of interest is the magnitude and phase of the function $T_s(j\phi)$. When proposing a design technique for digital filters which consider frequency domain properties it is usual to consider the filters to be equivalent if the Laplace (s) domain function $T_s(j\phi)$, and sampled (z) domain function $T_z(\cos[\phi t] + j\sin[\phi t])$ fulfill the same specifications.

The equivalence of these functions does not alter the fact that digital filter is sampled whereas the analog is not. Sampling introduces new frequency components, and these may alter the performance of the filter quite significantly. The complex function $T_z(\cos[\phi t]+j\sin[\phi t])$ refers to the magnitude and phase of the sinusoidal component at frequency ϕ. The introduced frequency components are not specified by this function.

The frequency domain design techniques presented here are based on the conventional assumption that the frequency response of a digital filter is specified by considering the sinusoidal component at the output. However the introduced frequency components can be very important since they are noise terms at the output of the digital filter. The significance of this noise in the given application must be checked during the filter design. The design techniques discussed here only consider the sinusoidal component at the output of the digital filter as specified by the z domain function $T_z(\cos[\phi t]+j\sin[\phi t])$.

2.1.3 Zero Order Hold

In many applications a zero order hold is present at the output of the digital filter. The sample and hold introduces significant amplitude and phase characteristics, yet few design procedures exist for this situation. Iterative techniques to adjust the transfer function to compensate for this magnitude characteristic have been devised [Antoniou et al. '82], and the introduction of a special second order section has been proposed [Henkel '68]. The design techniques presented here provide useful compensation quite conveniently.

2.1.4 Choice of Sampled Domain Operator

Some of the design techniques presented here are most clearly seen by introducing operators other than z to represent sampled functions.

Time domain functions are related to functions of s in the Laplace domain. Sampled time domain functions are related to functions of z. The choices of the operators s and z are made for convenience. Other choices can be made whilst preserving the fundamental properties of the transforms. For example it is straight-forward to consider a 1/s domain where pure integration has been changed from the conventional pole at the origin to a zero at the origin. Alternatively the 1/(s+1) domain could be defined for example, functions of s being changed to functions of 1/(s+1).

The Laplace operator s is a convenient choice because it is differentiation in the time domain. This suits many Laplace domain applications. For example the voltage to current relationships of capacitors and inductors are simple functions of s, and this simplifies circuit analysis.

Similarly the operator z directly relates to time domain functions. The coefficient of the term z^{-n} in the z domain expressions is the magnitude of the sample at $t = nT$. Consequently multiplication of a z domain function by z^{-1} corresponds to a delay in the time domain of one sampling interval.

The relationship, however, between the Laplace (s) operator and the sampled domain (z) operator is not always the most convenient. The operators differ considerably with s giving differentiation and z giving a time shift. Even at high sampling rates and with long time constants when the performances of digital and analog systems are very similar, the Laplace (s) and z domain representations differ considerably. For impulse invariance s domain singularities near the origin correspond to sampled (z) domain singularities near $z = +1$.

When considering equivalences between sampled and continuous systems it is sometimes helpful to reformulate the problem using more convenient operators. For example the digital differentiation operator

$$\beta T = z - 1 \qquad \ldots\ 2.1.1$$

has been proposed by Karwoski [Hackmeister '79].

The operator [Bolton '80a],

$$\gamma T = 1 - z^{-1} \qquad \ldots\ 2.1.2$$

was proposed to assist the analysis of digital filters. Both the β and γ operators approach the Laplace operator s at high sampling rates. In certain cases the operators β and γ enable similarities between sampled and continuous systems to be seen more readily. Some other aspects are most conveniently represented with the conventional z operator. Consider for example the proposed second order filter of figure 1.4 without feedforward terms. The low pass transfer function in the z domain is

$$T_{z,L}(z) = \frac{a^2z^2}{z^2-z(2-a^2-ab) + 1-ab} \qquad \ldots\ 2.1.3$$

in the 1/z

$$T_{1/z,L}(1/z) = \frac{a^2}{1 - (2-a^2-ab)z^{-1} + (1-ab)z^{-2}} \qquad \ldots\ 2.1.4$$

in the β

$$T_{\beta,L}(\beta) = \frac{a^2(\beta T+1)^2}{\{\beta T\}^2 + \beta Ta(1+ab) + a^2} \qquad \ldots\ 2.1.5$$

and in the γ domain

$$T_{\gamma,L}(\gamma) = \frac{a^2}{\{\gamma T\}^2(1-ab) + a\gamma T(b-a) + a^2} \qquad \ldots\ 2.1.6$$

Equations 2.1.3 to 2.1.6 represent exactly the same transfer function but certain properties are clearly evident in different domains. These properties include steady state gain and stability.

The proposed filter is all pole in the 1/z and in the y domains. It can be very convenient to represent the filter which has only feedback terms by an all pole transfer function. For example moving the singularities corresponds to a change in the coefficients, whereas changing the locations of the zeros of the transfer function in the z or β domains requires a change in the structure. Also, all pole transfer functions are most convenient to evaluate, a property which is used in the first frequency magnitude design technique presented here.

The steady state gain of the transfer function is unity. This is most clearly seen in the β and the y domains as the ratio of the constants in the numerator and denominator. The final value theorem in the Laplace domain gives the steady state response to a unit step as the value of the Laplace domain function as s approaches zero. In the z domain the steady state gain is obtained as z approaches unity. With both the β and y operators this corresponds to the operator approaching zero as is the case with the s operator. Therefore the steady state gain can be seen in the s, β and y domains simply by evaluating the constant terms.

For complex roots the transfer function of the proposed filter structure is marginally stable when the coefficient b = 0. This is evident most clearly in the z domain. The modulus of any complex z domain roots is unity whenever b = 0. The coefficient of z^0 in the quadratic for the roots is unity.

The sampled domains z, y and β all have circular frequency loci. In the β and y domains these loci pass through the origin, whereas in the z and 1/z domains the loci pass through +1. The frequency locus can be used to give an appreciation of the effects of sampling. Since the radius of the circle is proportional to the sampling frequency it is possible to estimate the effects of sampling by considering various radii. The y domain frequency locus was used to devise the frequency domain design techniques described in the following sections.

2.1.5 Zeros Introduced by the Bilinear Transform.

The usual design technique for digital filters using frequency domain equivalence is the bilinear transform. The substitution

$$z = \frac{1 + sT/2}{1 - sT/2} \qquad \text{... 2.1.7}$$

maps the frequency locus of the Laplace domain (jø axis) onto the frequency locus of the z domain (unit circle). When the transform of equation 2.1.7. is used the derived z domain function at $z = \cos[\phi]+j\sin[\phi]$ equals the Laplace domain function at $j\phi$.

The bilinear transform method is a very well established design technique. The particular property of interest here is that if this transform is applied to an all pole analog filter there are zeros to be implement in the z^{-1} or the y domain. These correspond to actual terms which must be implemented in the digital filter. For example the second order Chebyshev filter

$$T_s(s) = \frac{1}{\tau^2 s^2 + \tau s + 1} \qquad \text{... 2.1.8}$$

where the time constant/sampling interval, $\tau/T = 10$. Using the bilinear transform

$$sT = 2\,\frac{z - 1}{z + 1} \qquad \text{... 2.1.9}$$

gives the z domain transfer function

$$T_z(z) = \frac{0.002375\ (z + 1)^2}{z^2 - 1.8955z + 0.9050} \qquad \ldots\ 2.1.10$$

and the z^{-1} domain

$$T_{1/z}(z^{-1}) = \frac{0.002375(z^{-1} + 1)^2}{z^{-2} - 1.8955z^{-1} + 0.9050} \qquad \ldots\ 2.1.11$$

and the y domain

$$T_y(y) = \frac{(yT/2 - 1)^2}{95.25y^2T^2 + 9yT + 1} \qquad \ldots\ 2.1.12$$

It is clear that when the bilinear transform is used on an all pole analog transfer function feedforward terms are required for implementation. At large sampling rates the coefficients required to implement the feedforward terms are quite small. The contribution presented here is to show that it is possible to avoid the requirement to implement these feedforward terms by using alternative design procedures. Two new relationships between the sampled and the analog domains are given. The first provides an approximate frequency magnitude relationship whilst giving no frequency warping, whilst the second design procedure gives an exact frequency magnitude relationship but does give frequency warping.

2.2 Y Domain Frequency Magnitude Approximation

The relationship between the Laplace and the y domain is such that in the Laplace domain the frequency locus is the $j\phi$ axis whereas in the y domain it is a circle passing through the origin. The digital filter structure being considered implements an all pole y domain transfer function, and the property of interest is the frequency magnitude response.

An all pole y domain transfer function can be derived from an all pole Laplace domain transfer function simply by moving the poles. Similar frequency magnitude response can be obtained by locating the y domain poles so they are in the same relation to the same point of the frequency locus as they were in the Laplace domain. This is illustrated in figure 2.1.

Geometric relationships can be used to show that if the singularities in the Laplace domain are at $-a_s \pm jb_s$ those in the y plane are at $-a_y \pm jb_y$ where

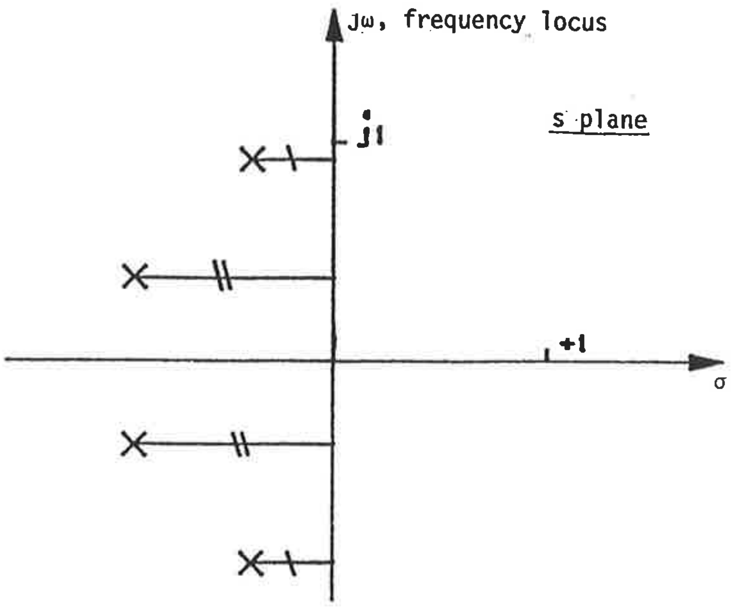
$$a_y = (1+a_s)\cos(b_s) - 1 \qquad \ldots\ 2.2.1$$

and

$$b_y = (1+a_s)\sin(b_s) \qquad \ldots\ 2.2.2$$

These simple relationships give a convenient design procedure.

A.



B.

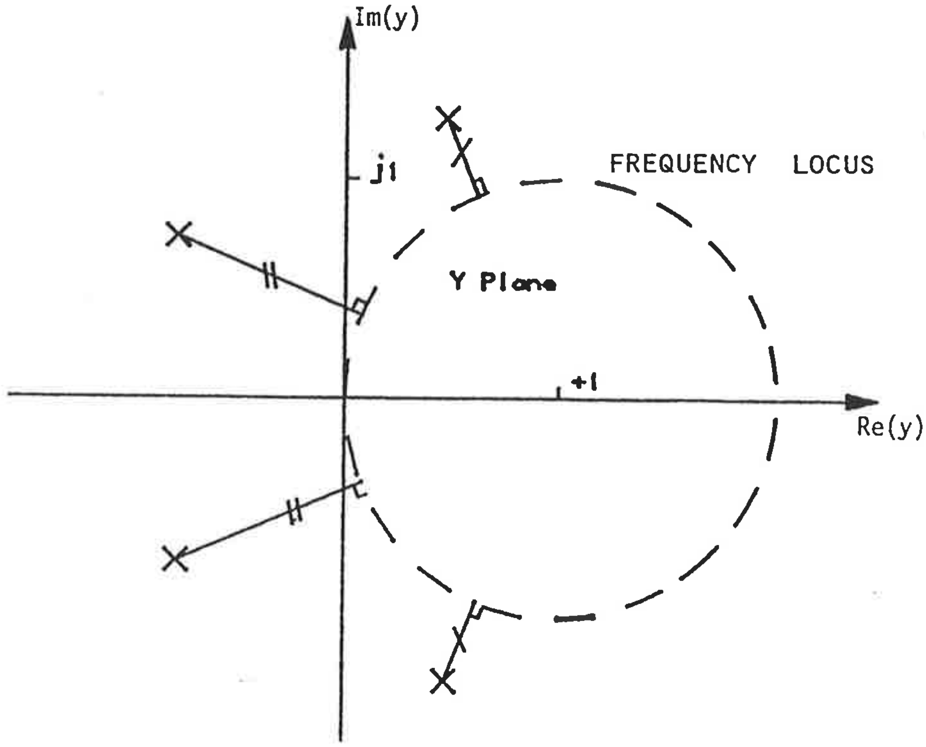


Figure 2.1 Movement of Singularities Between the s and the y Domains to preserve the relationship to the Frequency Locus.

At medium and high sampling rates this design procedure gives very useful agreement between the frequency responses of the analog and the compensated digital filters. Figure 2.2 compares the pass band responses between the analog and the derived digital filter for a fourth order Butterworth with a normalized cutoff frequency of 0.5. Table 2.A shows the maximum error within the pass band as a function of the normalized cutoff frequency for Butterworth filters.

The restriction on the usefulness of this design technique is the stop band performance rather than the pass band. Figure 2.3 compares the frequency magnitude response of the original analog filter and the derived digital filter. The stop band attenuation is not infinite because there are no zeros at $z = -1$, $(y = +2/T)$. A qualitative appreciation of the stop band attenuation can be obtained by considering the magnitude of the complex function at various points on the circular frequency locus in the y domain, as illustrated in figure 2.1.B. The stop band attenuation increases for high sampling rates and with higher order systems. In both these cases the proposed design technique is very useful.

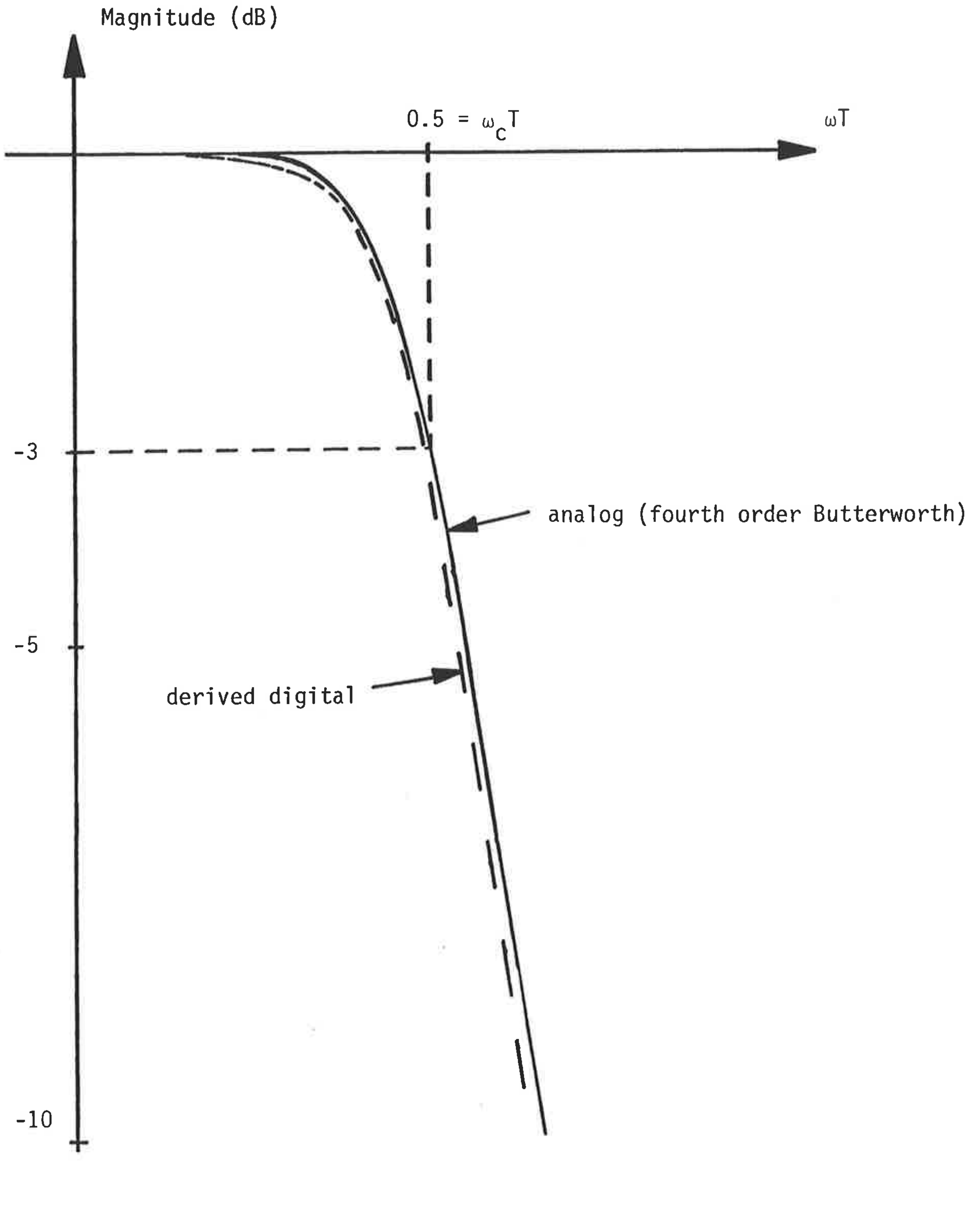


Figure 2.2 Responses of the Analog and Derived Digital Filter

Order of the Filter	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	0.21	0.41	0.59	0.76	0.91	1.06	1.19	1.31	1.42	1.52
2	0.01	0.04	0.09	0.15	0.23	0.33	0.45	0.58	0.73	0.90
3	0.10	0.16	0.21	0.22	0.21	0.18	0.12	0.03	0.09	0.23
4	0.15	0.25	0.32	0.34	0.33	0.28	0.20	0.08	0.09	0.28
5	0.19	0.33	0.42	0.47	0.47	0.42	0.32	0.18	0.01	0.25
6	0.24	0.41	0.53	0.58	0.58	0.53	0.42	0.25	0.03	0.25
7	0.28	0.48	0.62	0.69	0.70	0.64	0.51	0.32	0.07	0.26
8	0.32	0.56	0.72	0.80	0.81	0.75	0.60	0.39	0.10	0.27
9	0.36	0.63	0.82	0.91	0.92	0.85	0.69	0.45	0.13	0.29
10	0.40	0.71	0.91	1.02	1.03	0.95	0.78	0.51	0.15	0.30

Normalized Cutoff Frequency of the Butterworth Filter*

* $\phi_c T$, radians per second

Table 2.A Maximum Passband Error (dB) for the compensation of equations 2.2.1 and 2.2.2

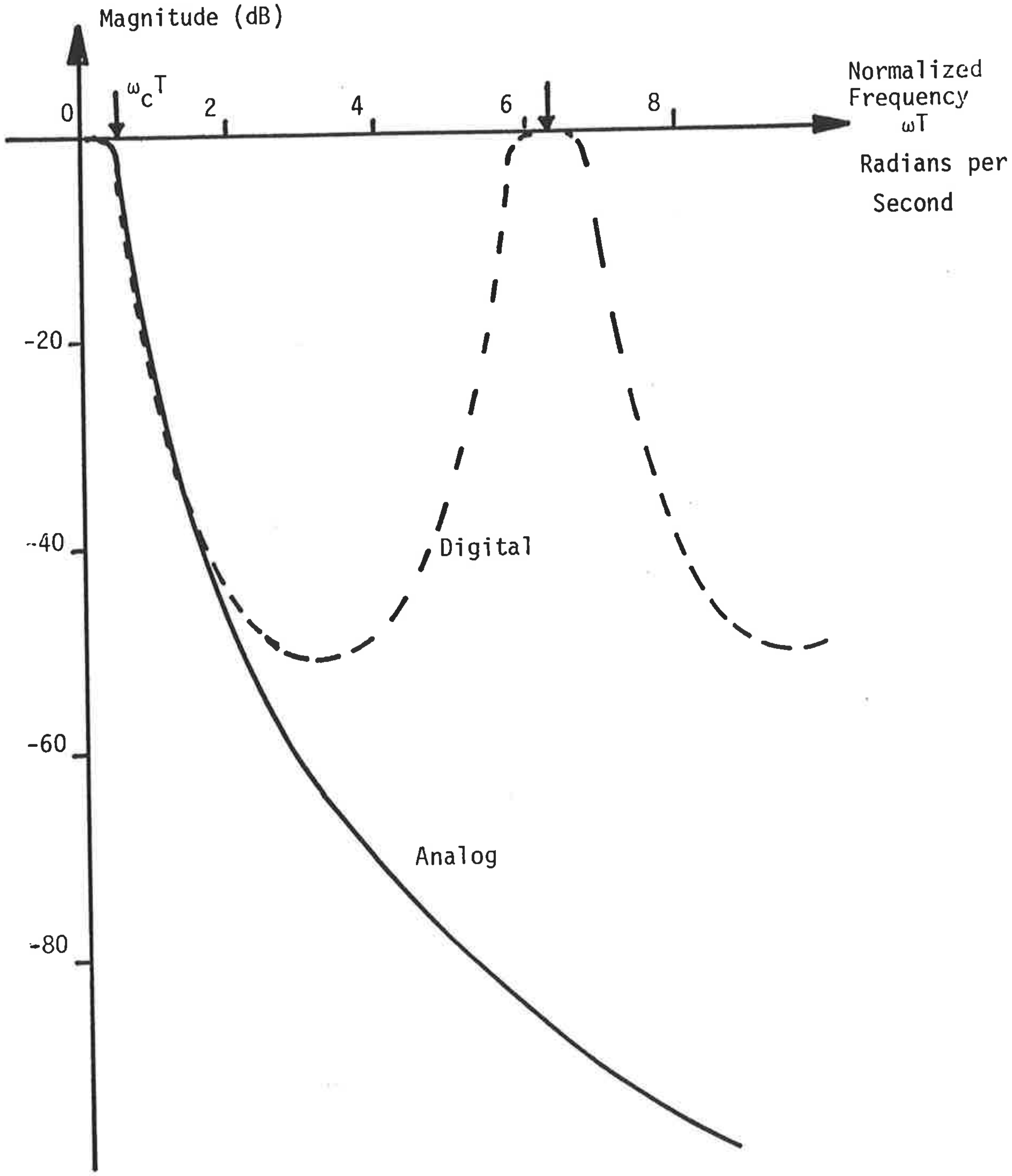


Figure 2.3 Frequency Response of the Filter Outside the Pass Band.

With lower sampling rates and with more stringent stop band attenuation specifications it becomes necessary to implement the zeros in the sampled domain. In this case the proposed digital filter structure is equally capable of implementing zeros which were obtained by applying the bilinear transform to an elliptic (Cauer) analog response. The z domain zeros are not all at $z = -1$, but are spaced around the unit circle to achieve equal stop band ripple.

The feature of this design technique is that there is no frequency warping, though the derived magnitude relationships are approximate. The advantage over the bilinear transform is that if harmonic or other relationships were present in the original analog filter transfer function, these would also be present in the derived digital filter. A further aspect is that the approximation using the relationship of poles or zeros to the frequency locus improves as the distance to the frequency locus becomes smaller. This means for very high Q singularities, either zeros or poles, the design technique is quite valuable.

2.2.1 Incorporating a Zero Order Hold

A zero order hold is often present at the output of the digital filter, so it is extremely important to have design techniques for this case. A zero order hold introduces a magnitude term at frequency ϕ of $\sin[\phi T]/\phi T$. Some compensation can be made in all pole filters by bringing the poles closer to the frequency locus. Relationships 2.2.1 and 2.2.2 become

$$a_y = (1+a_s\sin[b_s]/b_s)\cos[b_s] - 1 \qquad \ldots\ 2.2.3$$

and $$b_y = (1+a_s\sin[b_s]/b_s)\sin[b_s] \qquad \ldots\ 2.2.4$$

The results of these design relationships are given in Table 2.B. Although the compensation of equations 2.2.3 and 2.2.4 gives a considerable improvement, the agreement between the analog and derived digital responses can be poorer than if there were no zero order hold.

Order of the Filter										
1	0.22	0.46	0.72	0.99	1.28	1.59	1.91	2.26	2.63	3.02
2	0.00	0.02	0.04	0.08	0.12	0.17	0.24	0.32	0.41	0.51
3	0.10	0.20	0.29	0.37	0.44	0.51	0.57	0.62	0.67	0.71
4	0.15	0.28	0.38	0.46	0.52	0.56	0.57	0.56	0.54	0.49
5	0.20	0.35	0.47	0.56	0.60	0.61	0.59	0.53	0.44	0.32
6	0.24	0.42	0.56	0.64	0.68	0.66	0.60	0.49	0.33	0.12
7	0.28	0.49	0.64	0.72	0.74	0.70	0.60	0.43	0.21	0.09
8	0.32	0.56	0.72	0.80	0.81	0.74	0.60	0.38	0.08	0.29
9	0.36	0.63	0.80	0.88	0.87	0.78	0.59	0.32	0.05	0.50
10	0.40	0.69	0.88	0.96	0.94	0.81	0.59	0.26	0.17	0.71
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
	Normalized Cutoff Frequency of the Butterworth Filter*									

* $\phi_c T$, radians per second

Table 2.B Maximum Passband Error (dB) for the compensation of equations 2.2.3 and 2.2.4

2.2.2 Y Domain Frequency Phase Approximation

Often it is important to consider the phase response of a filter. The design technique presented in the previous section considered the frequency magnitude response only. The phase response was not considered at all. Indeed the phase response of the derived digital filters do differ considerably from the analog counter parts. In qualitative terms this means the waveforms of the output of the digital filter also differ considerably from the analog, even though the frequency magnitude response is equivalent.

Techniques for the design of all pole digital filters which are based on the filter's phase properties have been developed [Bolton '81c]. Second order sections were considered. The first and second differential of the phase response at arbitrarily low frequencies of the derived digital filter were set equal to the corresponding responses for the original analog filter. It was found that the frequency magnitude responses did differ considerably, as did the time responses. The reason is that the asymptotic relationships for arbitrarily low frequencies are not the best means for equivalence when the frequencies of interest are non-zero. For example if the overshoot in the step response is considered the sinusoidal components near the damped natural frequency of the pole pair will have the most effect. Therefore the design technique for preserving the phase relationships is not considered further here. New design techniques for the Bessel response for digital filters is presented later. The Bessel criterion considers all the available derivatives in an N^{th} order system, and the responses are quite useful, [Bolton '84d].

2.3 Magnitude Invariant Frequency Domain

Again the transform [Bolton '84a] between the Laplace and the z domain is designed to transform an all pole s domain filter to an all pole z^{-1} domain filter which does not require feedforward terms for implementation.

The bilinear transform

$$sT = 2 \frac{z - 1}{z + 1} \qquad \text{... 2.3.1}$$

has the inverse relationship

$$z^{-1} = \frac{1 - sT/2}{1 + sT/2} \qquad \text{... 2.3.2}$$

When this inverse relationship applied to an N^{th} order all pole z^{-1} domain function, a Laplace domain function is obtained which has N zeros at $s = -2/T$. A Laplace domain expression with one zero at $s = -2/T$ for each pole can be implemented by a digital filter without feedforward terms. This design uses the conventional bilinear transform with its frequency warping, but which otherwise does preserve the magnitude and phase relationships without approximation.

The contribution here is to show that it is possible to transform an all pole Laplace domain function to another Laplace domain function which has N zeros at $s = -2/T$. The frequency magnitude (but not phase) characteristic is preserved, and frequency warping does occur. This new transformation with the conventional bilinear transform enables an all pole Laplace domain expression to be implemented by a digital filter without feedforward terms whilst preserving the frequency magnitude characteristic. Again frequency warping does occur.

The design procedure is extended to consider compensation for a zero order hold at the output of the digital filter.

2.3.1 Derivation

A transfer function with single Laplace domain pole at $-\sigma/T$

$$H(s) = \frac{1}{sT/\sigma + 1} \qquad \ldots\ 2.3.3$$

can be transformed to a function $H'(s')$ of the variable s' whilst preserving the frequency magnitude properties. The square of the magnitude of $H(j\phi)$ is

$$[H(j\phi)][H(j\phi)]^* = \frac{1}{(\phi T/\sigma)^2 + 1} \qquad \ldots\ 2.3.4$$

where * denotes conjugation. Consider a new frequency variable ϕ' where

$$(\phi' T)^2 = \frac{(\phi T)^2}{1 - (\phi T/2)^2} \qquad \ldots\ 2.3.5$$

so that

$$(\phi T)^2 = \frac{(\phi' T)^2}{1 + (\phi' T/2)^2} \qquad \ldots\ 2.3.6$$

Substituting equation 2.3.6 into equation 2.3.4 gives the transfer function $H(j\phi)$ in terms of the new frequency variable ϕ'.

$$[H(j\phi' T)][H(j\phi' T)]^* = \frac{(\phi' T/2)^2 + 1}{(\phi' T/\sigma')^2 + 1} \qquad \ldots\ 2.3.7$$

where $(T/\sigma')^2 = (T/\sigma)^2 + (T/2)^2$ ... 2.3.8

Equation 2.3.7 for the square of the magnitude has a corresponding Laplace domain function

$$H'(s' T) = \frac{1 + s' T/2}{1 + s' T/\sigma'} \qquad \ldots\ 2.3.9$$

In the s' domain a zero is now present at $s'T = -2$ as required. The derivation for a single real pole is quite readily extended to general all pole transfer functions. When the transfer function is the product of several first order sections each section can be transformed to the s' domain using the given procedure [Bolton '84a].

When complex pole pairs are considered the variable σ of the above derivation is complex and the product of this pole with its conjugate is considered. Equation 2.3.9 can be applied to a complex pole at $s = -\sigma/T$ where σ is now complex.

The design procedures are simplified using relationships between the coefficients of the quadratics for σ and σ'. If σ is a solution to

$$a[sT]^2 + bsT + 1 = 0 \qquad \ldots\ 2.3.10$$

and σ' is a solution to

$$a'[s'T]^2 + b's'T + 1 = 0 \qquad \ldots\ 2.3.11$$

then it can be shown

$$a'^2 = a^2 + (b^2 - 2a)/4 + 1/16 \qquad \ldots\ 2.3.12$$

and

$$b'^2 = b^2 + 2(a' - a) + 1/2 \qquad \ldots\ 2.3.13$$

These are very useful design relationships which are convenient to implement [Bolton '84a].

2.3.2 Frequency Warping Characteristics

When the bilinear transform is applied to an all pole Laplace domain transfer function the frequency warping from the digital $\phi'T$ domain to the analog ϕT is

$$\phi T = 2\tan(\phi' T/2) \qquad \ldots\ 2.3.14$$

An infinite analog frequency, and hence infinite attenuation, transforms to $\phi'T = \Pi$ at the zeros where $z = -1$.

The combination of the analog transform between the s and the s' domains and then the bilinear transform gives a frequency characteristic.

$$\phi T = 2\sin(\phi' T/2) \qquad \ldots\ 2.3.15$$

A finite frequency ($\phi T = 2$) and hence finite attenuation is transformed to $\phi'T = \Pi$. Figure 2.4 shows the attenuation for the original analog filter, the filter obtained using the bilinear transform technique, and the filter obtained using the proposed all pole transform. The second order Butterworth example is chosen. It can be seen that whereas the frequency warping of the bilinear transform makes the attenuation change more rapidly with frequency the all pole transform reduces this rate of attenuation. It becomes important to consider frequency warping when choosing an analog transfer function to meet a specification.

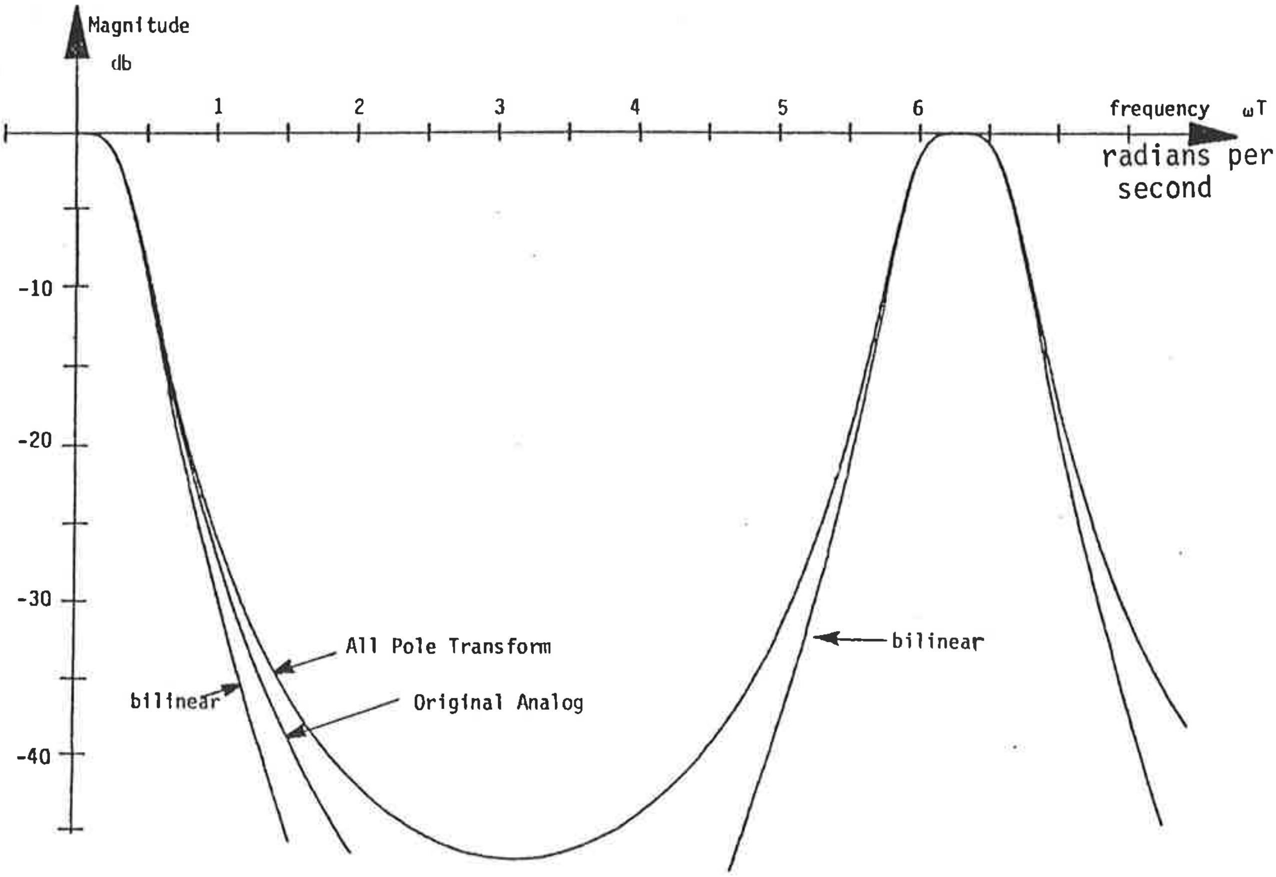


Figure 2.4 **Frequency Responses for Analog, Bilinear Digital and Magnitude Invariant Digital Filters.**

2.3.3 Incorporating a Zero Order Hold

A zero order hold is often used at the output of the digital filter and it introduces a frequency magnitude characteristic

$$|H(\phi T)| = \sin(\phi T)/(\phi T) \quad \text{... 2.3.16}$$

With this design technique it is possible to provide some compensation for this attenuation without additional hardware by locating the zeros of the derived analog transfer function at $s = -\Gamma$ rather than at $s = -2T$. This introduces a lead compensator $H'(s)$ where

$$H'(s) = \left[\frac{(sT/\Gamma) + 1}{(sT/2) + 1}\right]^N \quad \text{... 2.3.17}$$

where N is the order of the filter. The design relationships become

$$\phi T = \frac{2\Gamma\sin(\phi' T/2)}{\sqrt{[\Gamma^2\cos^2(\phi' T/2)+4\sin^2(\phi' T/2)]^2}} \quad \text{... 2.3.18}$$

Equation 2.3.8 is replaced by

$$(T/\sigma')^2 = (T/\sigma)^2 + 1/\Gamma^2 \quad \text{... 2.3.19}$$

Equations 2.3.12 and 2.3.13 are replaced by

$$a'^2 = a^2 + (b^2 - 2a)/\Gamma^2 + 1/\Gamma^4 \quad \text{... 2.3.20}$$

and $$b'^2 = b^2 + 2(a' - a) + 2/\Gamma^2 \quad \text{... 2.3.21}$$

The compensation is not exact. If the frequency for exact equivalence is $\phi_e T$ then

$$\sin(\phi_e)/\phi_e = \frac{(\phi_e T/2)^2+1}{(\phi_e T/\Gamma)^2+1} \qquad \ldots\ 2.3.22$$

This means the location of the zeros of the derived analog transfer function is given by

$$\Gamma = \frac{\phi_e T}{[\{(\phi_e T/2)^2+1\}\{\phi_e T/\sin(\phi_e T)\}^{2/N}-1]^2} \qquad \ldots\ 2.3.23$$

It is possible to show that for values of $\phi_e T$ between 0 and Π there is only one value of $\phi_e T$ for which equation 2.3.23 holds. Below this frequency the effect of the lead compensator exceeds that of the sample and hold so the magnitude of the compensated transfer function exceeds the magnitude of $H(j\phi'T)$. Above this frequency the attenuation of the $\sin(\phi T)/(\phi T)$ term predominates, aiding the stop band performance of low pass filters.

When designing low pass filters it is useful to choose the frequency of magnitude equality $\phi_e T$ by considering the effect of the compensation on the magnitude response for frequencies within the pass band. For frequencies below $\phi_e T$ the magnitude of the compensated transfer function will always be greater than the given analog function, whereas for frequencies between $\phi_e T$ and $\phi_c T$ it will be less. It is possible to choose $\phi_e T$ in relation to $\phi_c T$ so that the maximum positive gain error equals the maximum negative gain error within the pass band. The optimum proportion k in

$$\phi_e T = k \phi_c T \qquad \ldots\ 2.3.24$$

depends on the order of the filter N and the cutoff frequency $\phi_c T$. In most cases k = 0.9 gives nearly optimum results. Figure 2.7 shows the maximum excursion within the pass band as a function of N and $\phi_c T$. Up to $\phi_c T = 1$ it does not exceed 0.1 dB.

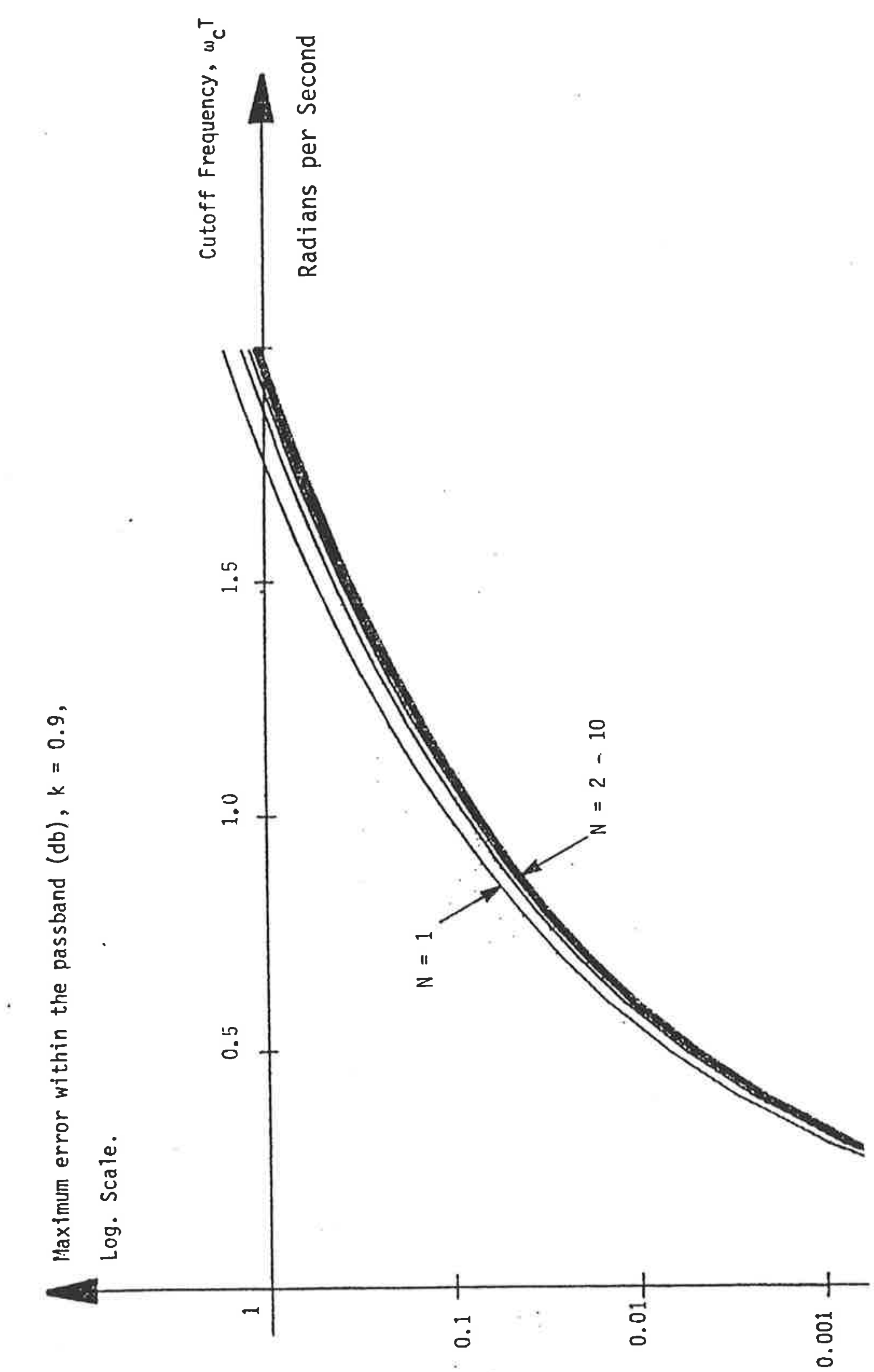


Figure 2.5 **Maximum Error within the Pass Band for the Zero Order Hold Compensation Technique.**

2.4 Unchanged Coefficients

The proposed digital filter structure of figure 1.4 has a very useful property. It is possible to completely ignore the effects of sampling by leaving the coefficients unchanged at their analog values and yet obtain an excellent approximation to the analog filter's frequency magnitude response, [Bolton '81c].

It has been seen that the Euler digital integration of the digital filter has replaced the analog integration of the original filter. Also there is one intrinsic delay in the feed back path which also changes the behaviour of the digital filter. The valuable property is that these two effects tend to cancel. In fact with absolutely no damping at all, coefficient $b = 0$, the derived digital filter is also marginally stable.

The cancellation is, of course, not complete. For example the frequency of the marginal stability of the digital filter does not exactly equal that of the original analog. Also, when damping is introduced the Qs of the filters are not identical either. However the responses are usefully similar, particularly when the convenience of the design procedure is considered. Table 2.C gives the error in the frequency magnitude response of the derived digital filter for various Butterworth responses.

However the responses do differ if a zero order hold is present at the output of the digital filter, principally because of the response of the zero order hold. Also, the time domain responses do differ, with the derived digital low pass filters having more overshoot than the original analog.

The intrinsic similarity in the frequency domain magnitude responses does mean that for less critical designs or for those with very large sampling rates, a suitable digital filter will be found by leaving the coefficients unchanged. It is quite simple to check the suitability of the filter using an evaluation of its response.

Order of the Filter	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	0.21	0.41	0.59	0.76	0.91	1.06	1.19	1.31	1.42	1.52
2	0.00	0.01	0.03	0.04	0.06	0.07	0.09	0.10	0.11	0.11
3	0.21	0.39	0.57	0.72	0.87	1.01	1.14	1.27	1.40	1.52
4	0.01	0.02	0.05	0.07	0.10	0.11	0.12	0.11	0.09	0.05
5	0.20	0.38	0.55	0.70	0.84	0.98	1.12	1.28	1.44	1.62
6	0.01	0.03	0.07	0.10	0.12	0.13	0.12	0.09	0.03	0.07
7	0.20	0.37	0.53	0.67	0.81	0.96	1.13	1.32	1.53	1.77
8	0.01	0.04	0.08	0.12	0.14	0.14	0.11	0.04	0.08	0.25
9	0.20	0.36	0.51	0.65	0.80	0.96	1.15	1.38	1.65	1.96
10	0.02	0.05	0.10	0.14	0.16	0.15	0.09	0.03	0.21	0.45

Normalized Cutoff Frequency of the Butterworth Filter*

* $\phi_c T$, radians per second

Table 2.C Maximum Passband Error (dB) case when the Coefficients of the Proposed Structure of figure 1.4 are Unchanged.

2.5 Bessel Digital Filters

The design of a digital filter given a satisfactory analog equivalent relies on the selection of a particular property of interest. The design procedures presented earlier gave digital filters which have a frequency magnitude response equivalent to a given analog filter. In general other properties and in particular the time domain properties will differ. These design procedures cannot be used if the original analog filter fulfilled the specification because of its step or impulse response.

Design procedures for impulse invariance or for step response invariance are very well established. The salient features are that if zeros are present in the original analog filter these will be moved in general, and, in particular, moved off the frequency locus. Hence the frequency domain response of analog filters with zeros will differ from the digital filter which has been derived using impulse invariance. The notch depths of the derived filters will be finite.

A second aspect of the impulse invariant design procedure is that when an all pole analog filter is transformed to a digital filter the zeros at infinity are transformed to finite zeros. Hence there is the requirement to implement feedforward terms in the digital implementation. There is the requirement for a new design procedure which minimizes the distortion of the pass band signal and gives an all pole y (or z^{-1}) domain digital transfer function.

In practice this requirement is quite important. There are many filter applications of interest where the time domain properties of the filter are very important. Filters derived using equivalence in the magnitude of the frequency response usually suffer a degradation of the time domain properties.

The new design procedure presented here is for the Bessel response [Bolton '84d]. This is a particularly important response when the time domain performance criteria are considered. The derivation involves deriving the Bessel criterion of maximally flat group delay for digital filters. The objective is to derive the all pole z^{-1} domain function of chosen order. The Bessel filter has as many of the derivatives of the group delay set to zero as the order of the filter allows, so the group delay is maximally flat.

2.5.1 Derivation

The transfer function of a digital filter with only feedback terms is

$$T_z(z) = \frac{1}{a_0 z^{-1} + a_1 z^{-2} + a_2 z^{-3} + \ldots} \quad \ldots\ 2.5.1$$

$$= C(z)/R(z) \quad \ldots\ 2.5.2$$

where C(z) and R(z) are the output and input functions respectively. The coefficient $a_0 = 1$ for canonic implementations of figure 2.6. This function is all pole in the z^{-1} domain. Then the filter can be implemented by cascading second order sections which are all pole in the z^{-1} domain. Being a linear system, when the input function is a sampled sinusoid the output is a sampled sinusoid of the same frequency, though in general the magnitude and phase will be changed.

Let the output be

$$c(nT) = cis(\phi nT) = \cos(\phi nT) + j\sin(\phi nT) \quad \ldots\ 2.5.3$$

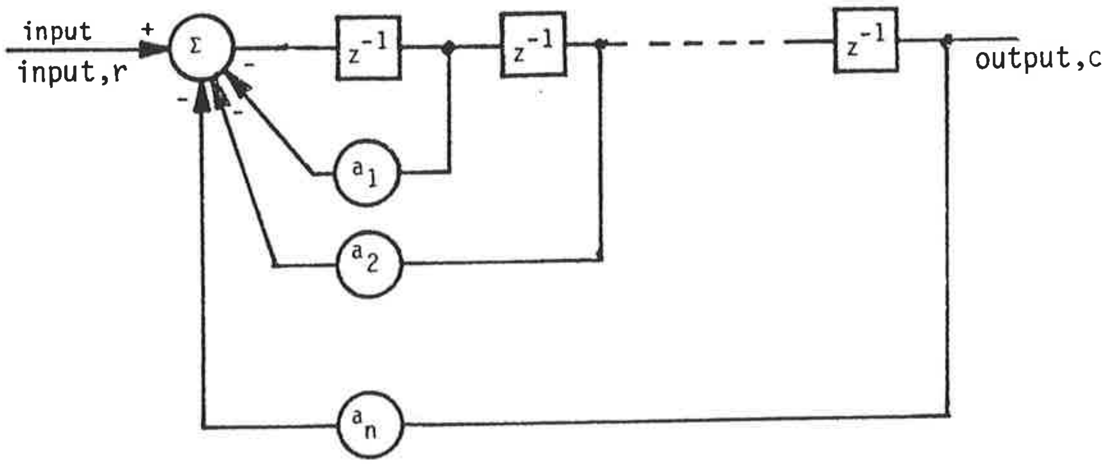


Figure 2.6 **Canonic Structure used in the Bessel Design Procedure.**

This means the input can be expressed using a magnitude function of frequency, $M(\phi)$, and a phase function of frequency, angle$[A(\phi)]$, as

$$r(nT) = M(\phi)\ \text{angle}[A(\phi)]\text{cis}(\phi T\{n+k\}) \quad \ldots\ 2.5.4$$

k is the desired delay of the Bessel response as a number of samples. In this case a Bessel response is obtained when both the phase function angle$[A(\phi)]$ at $\phi = 0$ is zero and its derivatives are zero up to the highest order that is possible given the order of the filter.

The summer of the structure of figure 2.6 ensures that at time $t = nT$

$$c(nT) = r(nT) - \sum_{i=1}^{n} [c(\{n-i\}T)a_i] \quad \ldots\ 2.5.5$$

where a_i is the i^{th} coefficient of the filter. Equation 2.5.5 gives

$$r(nT) = c(nT) \sum_{i=0}^{n} [a_i q^{-i}] \quad \ldots\ 2.5.6$$

where again the coefficient $a_0 = 1$ and q^{-1} is the delay operator which is defined so that $q^{-n}f(t) = f(t-nT)$.

Substituting in equation 2.5.6 for r(nT) and c(nT) using equations 2.5.3 and 2.5.4 respectively gives

$$M(\phi)\text{angle}[A(\phi)]\text{cis}(\phi T\{n+k\}) = \sum_{i=0}^{n} [a_i \text{cis}(\phi T\{n-i\})] \quad \ldots\ 2.5.7$$

Canceling the cis(øT{n+k}) term and rearranging gives

$$M(\phi)\text{angle}[A(\phi)] = \sum_{i=0}^{n} a_i \text{cis}(-\phi T\{i+k\}) \quad \ldots 2.5.8$$

Differentiating M(ø)angle[A(ø)] with respect to øT gives

$$d\{M(\phi)\text{angle}[A(\phi)]\}/dt = \sum_{i=0}^{n} -j(i+k)a_i \text{cis}(-\phi T\{i+k\}) \quad \ldots 2.5.9$$

$$d\{M(\phi)\text{angle}[A(\phi)]\}/dt = \sum_{i=0}^{n} -j(i+k)a_i, \text{ as } \phi T \text{ approaches } 0$$

$$\ldots 2.5.10$$

Similarly for the p^{th} derivative

$$d^pM(\phi)\text{angle}[A(\phi)]/[d\phi T]^p = \sum_{i=0}^{n} (-j)^p(i+k)^p a_i \text{cis}(-\phi T\{i+k\}) \quad \ldots 2.5.11$$

The derivatives of odd order are imaginary as ø approaches 0 whereas those for even order are real. Since angle[A(ø)] approaches 0 as ø approaches 0, only the imaginary derivatives contribute to the group delay.

Therefore the equations for the Bessel coefficients are

$$\sum_{i=0}^{n} (i+k)^p a_i = 0 \text{ , for } p = 1, 3, 5, \ldots 2n-1 \quad \ldots 2.5.12$$

Letting $a_0 = 1$ to suit the canonic implementation equation 2.5.12 gives

$$\sum_{i=1}^{n} (i+k)^p a_i + k^p = 0 \text{ for } p = 1, 3, 5, .. 2n-1 \qquad ... 2.5.13$$

The filter can be implemented using the proposed filter structure in which case a constant factor is introduced into the transfer function which normalizes the steady state gain of the filter.

2.5.2 Design Procedure

The design procedure involves solving for the coefficients a_i of equation 2.5.13 given the order of the filter, N, and the desired group delay as k, the number of samples. Define matrix G so that its elements

$$g_{ij} = (i+k)^{2j-1} \qquad ... 2.5.14$$

and matrix H with elements

$$h_i = k^{2i-1} \qquad ... 2.5.15$$

and matrix A such that its elements a_i are the coefficients of the canonic filter. Then equation 2.5.15 becomes

$$A = -G^{-1}H \qquad ... 2.5.16$$

The procedure is to set up the matrices G and H and solve for A which gives the coefficients of the canonic filter.

2.5.3 Results

Figure 2.7.A compares the step responses for two second order Bessel filters. Both have the same time delay, 2 units, but one has one sample per unit time whereas the second has 10 samples. Figure 2.7.B repeats these responses for a ninth order filter. The maximally flat group delay property provides a very similar overshoot in the step response in all cases, approximately 0.4%. The responses are qualitatively similar, though as the sampling rate is decreased there are high frequency terms in the output as indicated by the discontinuity at the beginning of the response.

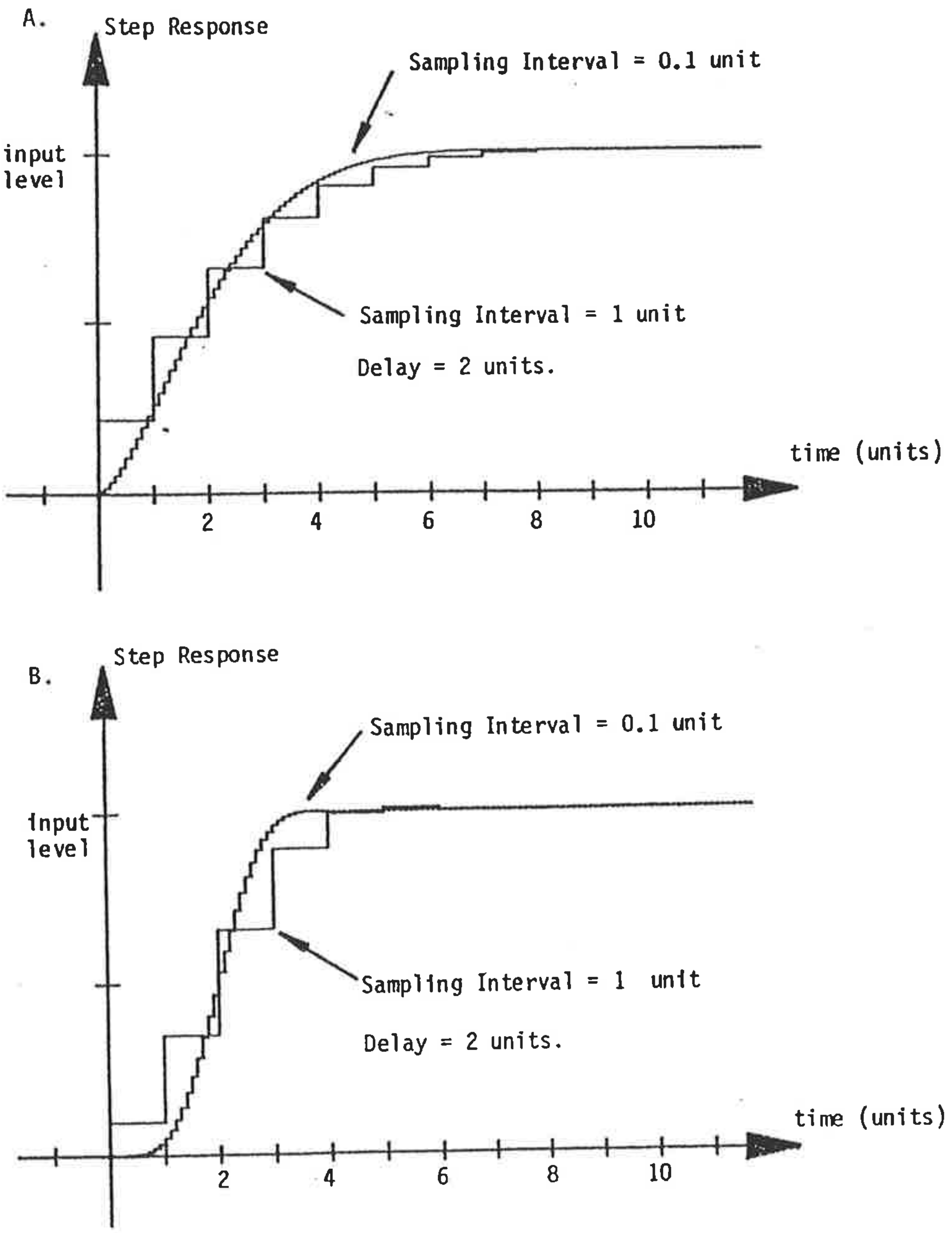


Figure 2.7 Comparison of Step Responses for the Bessel Filters

(A) Second Order

(B) Ninth Order

2.6 Summary.

The design procedures which have been presented here were developed considering the convenience of their implementation. It has been seen that the procedures which consider frequency domain properties can be implemented as simple relationships between coefficients of quadratic polynomials. The Bessel property of a response with a maximally flat group delay can be implemented by solving of a polynomial which is of the order of the filter.

These new design procedures have been developed to augment the conventional range of available design techniques. The bilinear transform gives the most appropriate design technique when the original analog filter has finite zeros of transmission. The magnitude invariant transform can be used when the original transfer function is all pole in the Laplace domain and the sampling rate is high in relation to the cutoff frequency. In applications where distortion to the signal waveform is important the direct derivation of the Bessel response can be used. In this way the new techniques presented here allow a greater selection in the choice of digital filter designs.

Chapter 3 NONLINEAR CONSIDERATIONS

3.1 Introduction

The previous chapters contain a description of a versatile second order recursive digital filter section. Design procedures for the coefficients were given. These procedures were derived using the assumption that the filter's behaviour is linear. In fact the filter has registers of finite length. In consequence the arithmetic operations are nonlinear. This chapter contains design procedures for additional program steps to minimize the effects of the corresponding nonlinear properties.

The objective of the analysis presented in this chapter is to obtain program steps which prevent oscillations and steady state errors. A feature of the analysis is that non-zero inputs are considered. Classen and Kristiansson [Classen 75] have given bounds for the stability of the canonic second order recursive digital filter, which has one non-linearity. The technique used by Classen is Tsypkin's stability criterion, [Tsypkin 65], which can be used when there is one non-linear element. In filter considered here there are five points throughout the filter where non-linearities may occur. The technique used here is based on the Liapunov, [Kuo 80]. Barnes and Fam have showed that autonomous limit cycles are not sustained in the minimum norm structure, [Barnes and Fam '80]. The proof applies to all bounded nonlinearities, including the two's complement overflow. Extensions to this work, [Bolton '84b], are included in this chapter.

The nonlinear properties of a digital filter depend on both the details of its structure as well as the nonlinearities of the arithmetic used for its implementation. An important aspect is that arithmetic nonlinearities occur during each evaluation within the filter, and that the filter program is executed sequentially. It follows the result of a nonlinearity is propagated to successive signal variables.

There are two nonlinear properties associated with two's complement arithmetic, signal overflow and quantization. These are illustrated in figure 3.1. Each of these nonlinearities can alter the response of the filter. Nonlinearities can give sustained steady state errors or zero input (autonomous) limit cycles as shown in figure 3.2. Saturation is the overflow characteristic considered here. Saturation can be obtained in microprocessor implementations of digital filters using tests for overflow conditions. The two's complement quantization is the small signal characteristic considered. The analyses presented here apply only to the filter section of the previous chapters, (Figure 3.3).

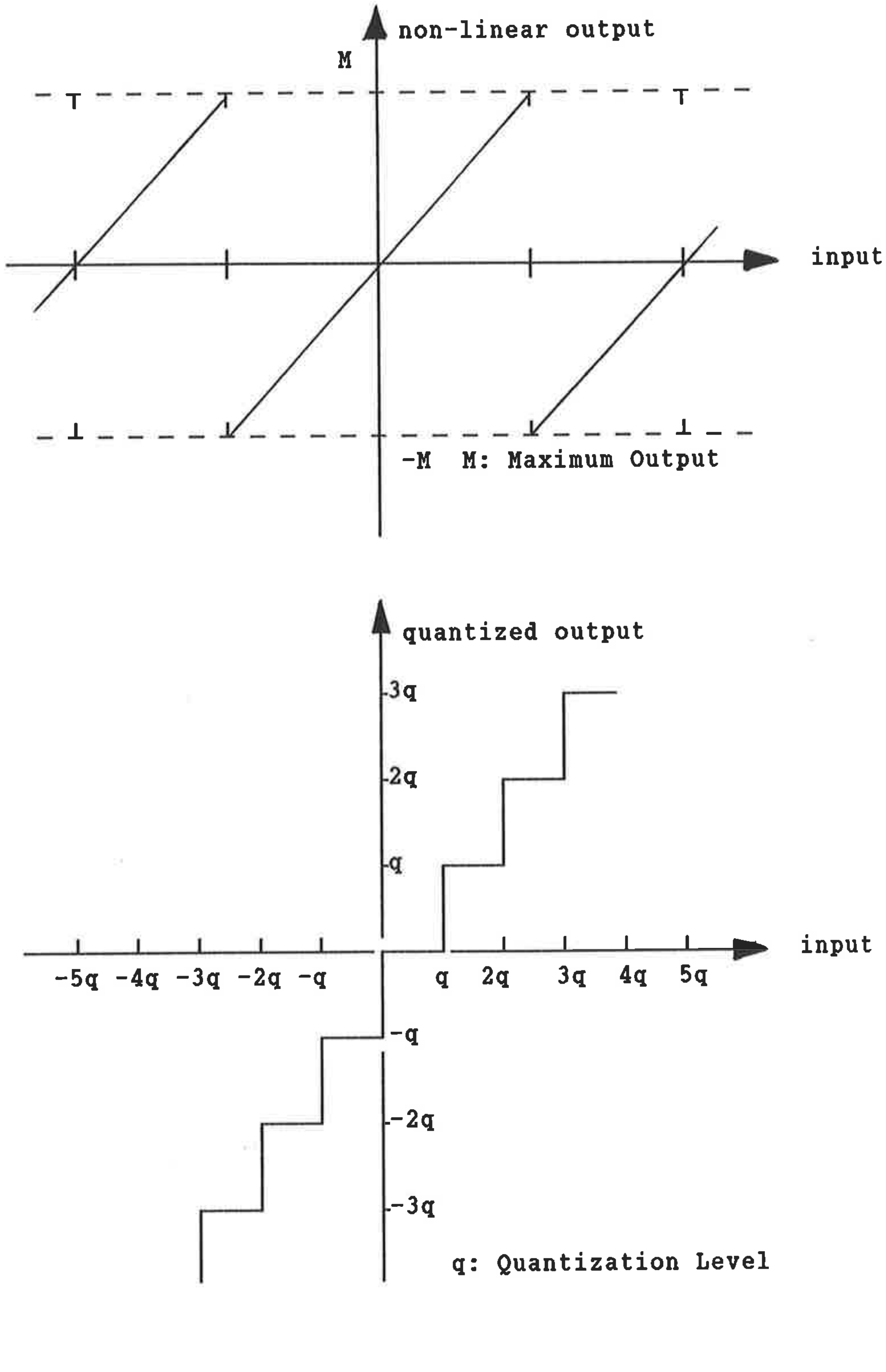


Figure 3.1 Two's complement overflow and quantization characteristic

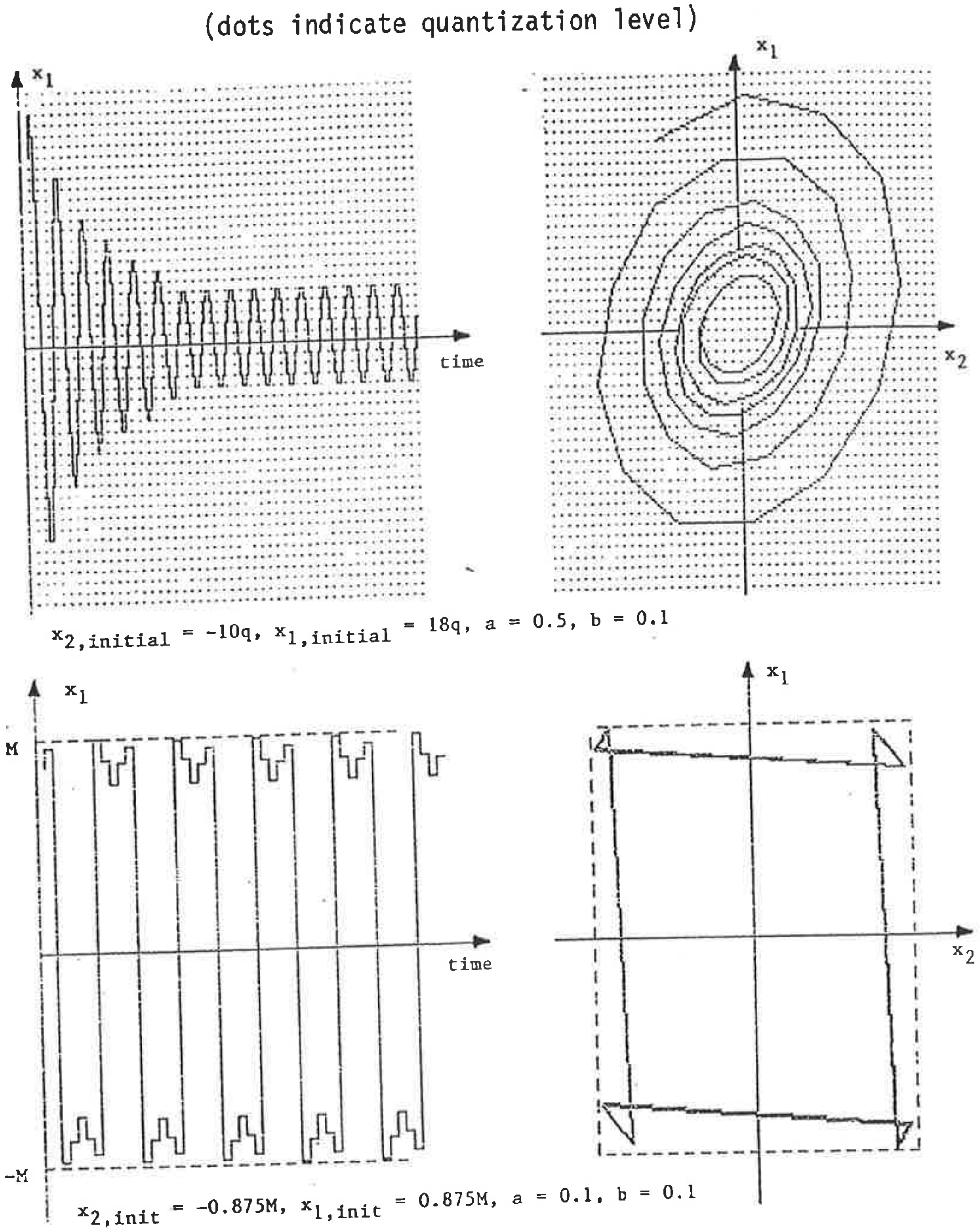


Figure 3.2 Time Domain and Phase Portrait Representations of Large and of Small Signal Limit Cycles.

3.1.1 Outline of Presentation

In section 3.2 the filter which is analysed is specified (equations 3.2.22 to 3.2.25), and a test function (equation 3.2.5) is considered for Liapunov properties.

In section 3.3 the zero input properties of the filters are addressed. It is shown that saturation overflows during the computation of x_1 and of x_2 cannot give limit cycles (equations 3.3.44 and 3.3.60). However no corresponding proof exists for overflows at x_3 and x_0 (equations 3.3.86 and 3.3.110).

Then, in section 3.4, the Liapunov analysis is extended to non-zero inputs. It is shown that saturation overflows in x_2 cannot give sustained oscillations for a range of coefficients given in equation 3.4.35. Equation 3.4.57 is the corresponding range of coefficients for saturation overflows in x_1.

With saturation overflows in x_0 and x_3 the test function can be used to find the non-zero inputs which give sustained oscillations. Figure 3.4 gives the result of such a search.

Finally, in section 3.5 Liapunov analysis is used to devise a procedure to guarantee the elimination of small signal autonomous limit cycles (program steps 3.5.22 to 3.5.28).

3.2 A Liapunov Function

The second method of Liapunov [Kuo '80] is useful for the study of nonlinear systems. The method states that if a function of the state variables, Lf(X), exists with the following four properties then the equilibrium state $x_k = 0$ for all k is asymptotically stable in the large, and Lf(X) is a Liapunov function.

i. $Lf[x_k = 0] = 0$ ... 3.2.1

The Liapunov function is zero at the equilibrium condition.

ii. $Lf[x_k <> 0] > 0$ ... 3.2.2

The Liapunov function is positive whenever the system is not at equilibrium.

iii. Lf[X] approaches infinity as ||X|| approaches infinity. ... 3.2.3

iv. $Lf[X(k+1)] < Lf[X(k)]$ for $X<>0$ and for all k ... 3.2.4

The method of Liapunov is a sufficient condition for stability. Hence, although the existence of a Liapunov function ensures global stability, a globally stable system may not have a Liapunov function. In general a difficulty in applying the method is to find functions which have the Liapunov properties. For the second order digital filter section there are a number of Liapunov functions of the form

$$Lf(x_1,x_2) = g_1x_1^2 + g_2x_2^2 + g_3x_1x_2 \qquad \ldots 3.2.5$$

where g_1, g_2 and g_3 are constants. A particularly useful Liapunov function is

$$Lf(x_{1,i},x_{2,i}) = (1-ab)x_{1,i}^2 + x_{2,i}^2 - (a-b)x_{1,i}x_{2,i} \qquad \ldots 3.2.6$$

It is now shown that this particular Liapunov function changes in a constant proportion at each computation loop of the filter. Since, for the filter of Figure 1.5

$$x_{1,i+1} = (1-ab)\ x_{1,i} - a\ x_{2,i} \qquad \ldots 3.2.7$$

$$x_{2,i+1} = a(1-ab)\ x_{1,i} + (1-a^2)\ x_{2,i} \qquad \ldots 3.2.8$$

From equations 3.2.6 and 3.2.7

$$Lf(x_{1,i+1},x_{2,i+1}) = x_{1,i+1}^2 + x_{2,i}(x_{2,i+1} + bx_{1,i+1}) \qquad \ldots 3.2.9$$

$$= [(1-ab)x_{1,i} - ax_{2,i}]^2 + x_{2,i}[a(1-ab)x_{1,i}$$
$$+ (1-a^2)x_{2,i} + b\{(1-ab)x_{1,i} - ax_{2,i}\}] \qquad \ldots 3.2.10$$

$$= (1-ab)^2x_{1,i}^2 - 2a(1-ab)x_{1,i}x_{2,i} + a(1-ab)x_{1,i}x_{2,i}$$
$$+ (1-ab)x_{2,i} + b(1-ab)x_{1,i}x_{2,i} \qquad \ldots 3.2.11$$

$$= (1-ab)[(1-ab)x_{1,i}^2 + x_{2,i}^2 - (a-b)x_{1,i}x_{2,i}] \qquad \ldots 3.2.12$$

$$= (1-ab)Lf(x_{1,i},x_{2,i}) \qquad \ldots 3.2.13$$

Since the filter's coefficients 'a' and 'b' are constant the factor (1-ab) is also constant. Hence this particular Liapunov function changes in a constant proportion at each computation loop.

This Liapunov function is positive for all values of x_1 and x_2 (positive definite) for certain values of the coefficients. It is only when these coefficients are within these limits that the Liapunov function exists. The bounds on the coefficients can be found.

$$Lf(x_1,x_2) = (1-ab)x_1^2 + x_2^2 - (a-b)x_1x_2 \qquad \ldots\ 3.2.14$$

$$= (1-ab-[(a-b)/2]^2)x_1^2 + (x_2-x_1(a-b)/2)^2 \qquad \ldots\ 3.2.15$$

The function is positive for all values of x_1 and x_2 (positive definite) for certain values of coefficients a and b. The bounds of the coefficients 'a' and 'b' for the function to be positive definite are found by ensuring the coefficient of the square term is positive. These bounds are

$$0 < 1-ab-[(a-b)/2]^2 \qquad \ldots\ 3.2.16$$

$$0 < 4 - 4ab -a^2 - b^2 + 2ab \qquad \ldots\ 3.2.17$$

$$0 < 4 - (a+b)^2 \qquad \ldots\ 3.2.18$$

$$2 > |a+b| \qquad \ldots\ 3.2.19$$

It will be shown that this is the condition for complex roots of the filter's transfer function, (equation 3.2.39).

If $|1-ab| < 1$ the Liapunov function decreases to zero the filter is stable. This is the condition which corresponds to the complex poles of the transfer function being within the unit circle. The feature of Liapunov analysis is that it can be applied to nonlinear systems. If it can be shown that nonlinearities can only reduce the Liapunov function then the stability of the nonlinear operation of the filter is guaranteed.

It can be useful to express this Liapunov function as

$$Lf(x_{1,i},x_{2,i}) = x_{1,i}^2 + (x_{2,i} - ax_{1,i})(x_{2,i} + bx_{1,i}) \quad \ldots 3.2.20$$

$$= x_{1,i}^2 + x_{2,i-1}(x_{2,i} + bx_{1,i}) \quad \ldots 3.2.21$$

The bounds on the coefficients are the same bounds which apply using the linear analysis of the digital filter section. The Liapunov analysis of the filter with linear operations has been applied in equations 3.2.6 to 3.2.13. In section 3.3 the Liapunov method is applied to the filter when its arithmetic operations are nonlinear. Section 3.4 applies Liapunov analysis to the systems with non-zero inputs by considering the transient component of the total response.

3.2.1 Program Steps of the Filter Section

When applying nonlinear analysis it is important to specify the sequence of instructions. The result of any nonlinearity in an evaluation will be present in the successive program steps.

The program steps for each computation loop of the filter are

$$x_0 = - bx_1 \quad \ldots 3.2.22$$

$$x_3 = x_0 - x_2 \quad \ldots 3.2.23$$

$$x_1 = x_1 + ax_3 \quad \ldots 3.2.24$$

$$x_2 = x_2 + ax_1 \quad \ldots 3.2.25$$

The variables x_1 and x_2 are re-computed at the program steps $x_1 = x_1+ax_3$ and $x_2 = x_2+ax_1$. Subscripts i and i+1 are used to analyze the filter. At the i^{th} computation the variables $x_{1,i}$ and $x_{2,i}$ are used to compute $x_{1,i+1}$ and $x_{2,i+1}$.

The computations of the program with an input of zero are

$$x_{0,i+1} = -bx_{1,i} \quad \ldots 3.2.26$$

$$x_{3,i+1} = x_{0,i+1} - x_{2,i} \quad \ldots 3.2.27$$

$$x_{1,i+1} = x_{1,i} + ax_{3,i+1} \quad \ldots 3.2.28$$

$$x_{2,i+1} = x_{2,i} + ax_{1,i+1} \quad \ldots 3.2.29$$

3.2.2 State Transition Matrix

$x_{1,i+1}$ and $x_{2,i+1}$ can be expressed directly in terms of $x_{1,i}$ and $x_{2,i}$.

$$x_{1,i+1} = (1-ab)\ x_{1,i} - a\ x_{2,i} \qquad \text{... 3.2.30}$$

$$x_{2,i+1} = a(1-ab)\ x_{1,i} + (1-a^2)\ x_{2,i} \qquad \text{... 3.2.31}$$

In matrix form these equations become

$$\begin{vmatrix} x_{1,i+1} \\ x_{2,i+1} \end{vmatrix} = \begin{vmatrix} (1-ab) & -a \\ a(1-ab) & (1-a^2) \end{vmatrix} \begin{vmatrix} x_{1,i} \\ x_{2,i} \end{vmatrix} \qquad \text{... 3.2.32}$$

There is a close relationship between this transition matrix and the corresponding Liapunov function. For complex poles the filter is stable if the determinant of the transition matrix has a magnitude which is less than unity. That is

$$|(1-ab)(1-a^2)+a^2(1-ab)| < 1 \qquad \text{... 3.2.33}$$

hence $$|1-ab| < 1 \qquad \text{... 3.2.34}$$

which is the necessary and sufficient condition for a reduction in the magnitude of the Liapunov function [3.2.13].

3.2.3 Transfer Function of the Filter Section

The block diagram of the second order digital filter considered here is given in figure 3.3. The corresponding transfer function is

$$T_z(z) = \frac{a^2z^2}{z^2 - z(2-ab-a^2) + 1-ab} \qquad \ldots\ 3.2.35$$

The z domain poles of the transfer function are given by the equation

$$z^2 - z(2-ab-a^2) + 1-ab = 0 \qquad \ldots\ 3.2.36$$

The coefficients of engineering interest are those which give stable transfer functions. Furthermore real poles are of little engineering interest because these can be implemented using cascaded first order stages. These conditions give bounds on the filter coefficients, 'a' and 'b'.

The necessary and sufficient condition for complex roots is that the determinant of the quadratic (equation 3.2.36) is negative.

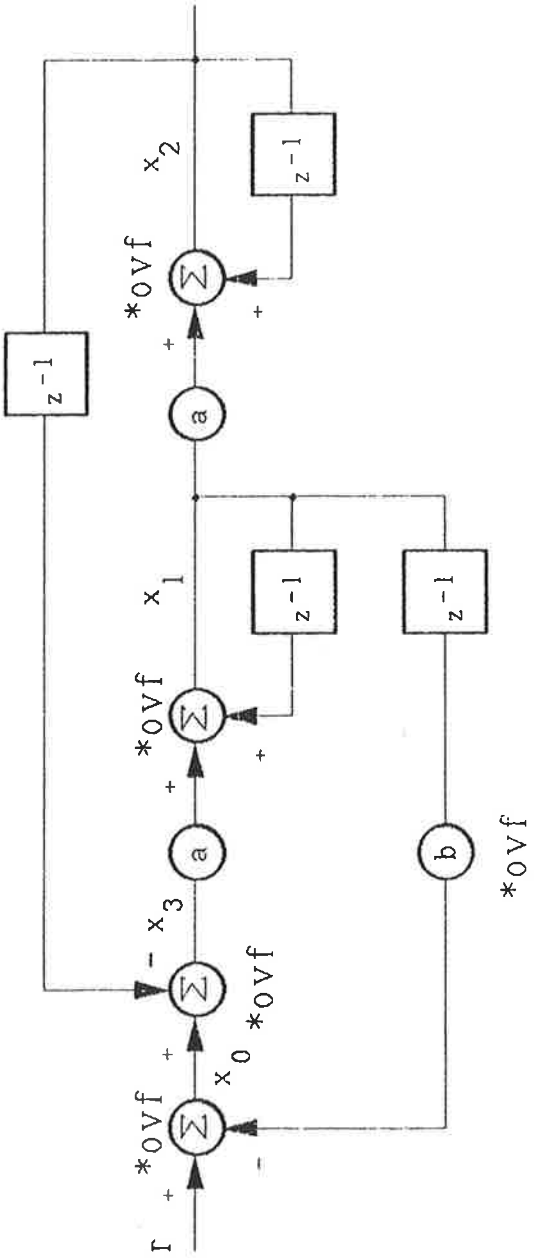


Figure 3.3 Block Diagram Representation of the Second Order Filter (section 3.2.1), Illustrating where Overflows May Occur.

This determinant is

$$[2-a(a+b)]^2 - 4(1-ab) < 0 \quad \ldots 3.2.37$$

$$a^2(a+b)^2 - 4a(a+b) + 4ab < 0 \quad \ldots 3.2.38$$

which gives the condition

$$(a + b)^2 < 4 \quad \ldots 3.2.39$$

This is the necessary condition (equation 3.2.19) for the Liapunov function to be positive definite.

The coefficient of z^0 in the characteristic equation of the transfer function of the filter (equation 3.3.35) is 1-ab. This coefficient is the product of the magnitudes of the two roots, so the necessary and sufficient condition for complex poles to be within the unit circle is

$$(1 - ab)^2 < 1 \quad \ldots 3.2.40$$

This is the necessary and sufficient condition for the Liapunov function to decrease monotonically. The corresponding property of the state transition matrix is given in equation 3.2.34.

From equation 3.2.39

$$a^2 + b^2 + 2ab < 4 \qquad \text{... 3.2.41}$$

but

$$a^2 + b^2 - 2ab > 0 \qquad \text{... 3.2.42}$$

subtracting 3.2.42 from 3.2.41 gives

$$ab < 1 \qquad \text{... 3.2.43}$$

conditions 3.2.40 and 3.2.43 give

$$0 < 1 - ab < 1 \qquad \text{... 3.2.44}$$

Which are the bounds of the coefficient values of the filter section which are of particular engineering interest.

3.3 Liapunov Analysis of Autonomous Limit Cycles

In practice if the digital filter is programmed using two's complement arithmetic, and there are no program steps to avoid the intrinsic two's complement numeric properties, overflows will arise in normal operation. Even so the filter almost always operates without displaying large signal limit cycles. However it is possible with care to choose initial conditions and coefficient values which do give rise to sustained zero-input oscillations. Once these oscillations have started they are difficult to remove. It is necessary to re-initialize the filter to resume normal operation. The possibility of large signal oscillations must be removed in an operational digital filter.

There is a range of coefficients 'a', and 'b', where the filter is stable provided it is operating without nonlinearities occurring. When the digital filter is stable and linear the Liapunov function decreases with each computation. The application of the Liapunov technique to a nonlinearity can involve a comparison of the Liapunov functions with and again without the nonlinearity. If the nonlinearity cannot increase the Liapunov function, then the analysis can guarantee the filter with its nonlinearity is stable. If the nonlinearity results in an increase in the Liapunov function, then the filter may be unstable. The conventional analysis applies to autonomous limit cycles, in which case the input is assumed to be zero. This assumption will be reviewed later.

The filters of engineering interest have z domain poles in the region of $z = +1$. This region corresponds to values of coefficient 'a' between zero and unity. Assuming this restriction on the range of coefficient 'a', there are five points throughout the filter where signal overflows may occur. Four of these points are the additions, and the fifth is a multiplication by coefficient 'b' which may be greater than unity. These points are labeled on the block diagram of figure 3.3.

The intrinsic two's complement overflow characteristic changes the signal by 2M, where +M and -M are the bounds of the signal. The saturation overflow reduces the magnitude of the signal so that its magnitude is M. The error term 'e' added to the signal by saturation may be quite small, and saturation does not change the sign of the signal.

An example of an overflow limit cycle is given in figure 3.2. The phase portrait of the filter, in this case a graph of the variable x_1 as a function of x_2, is shown.

3.3.1 Two's Complement Overflow in x_2

The function presented in section 3.2 for Liapunov analysis is

$$Lf(x_{1,i+1}, x_{2,i+1})$$

$$= (1-ab)x_{1,i+1}^2 + x_{2,i+1}^2 - (a-b)x_{1,i+1}x_{2,i+1} \quad \text{... 3.3.1}$$

$$= x_{1,i+1}^2 + (x_{2,i+1} - ax_{1,i+1})(x_{2,i+1} + bx_{1,i+1}) \quad \text{... 3.3.2}$$

$$= x_{1,i+1}^2 + x_{2,i}(x_{2,i+1} + bx_{1,i+1}) \quad \text{... 3.3.3}$$

This function was shown to be a Liapunov function when the filter operates in its linear region with coefficients so as to give a transfer function with stable complex singularities. The conditions a Liapunov function must satisfy include being positive definite, and its magnitude must decrease at each computation.

If the Liapunov properties of this function apply when the arithmetic exhibits a two's complement overflow in the variable $x_{2,i+1}$, then a Liapunov function exists and the nonlinear filter cannot be unstable. The function will be referred to as the "test function", $Lf(x_1, x_2)$, until the Liapunov properties are demonstrated. The property of particular interest is that the magnitude of the function must decrease at each computation.

When considering two's complement overflow in the computation of variable $x_{2,i+1}$ it is important that this low pass output of the filter is the last variable which is computed during each computation loop. Hence an arithmetic nonlinearity in the computation of this variable does not alter the values of the other variables computed during the same computation loop. At the i^{th} computation the variables $x_{1,i}$ and $x_{2,i}$ are used to compute $x_{1,i+1}$ and $x_{2,i+1}$ using the following computations.

$$x_{0,i+1} = - bx_{1,i} \qquad \text{... 3.3.4}$$

$$x_{3,i+1} = x_{0,i+1} - x_{2,i} \qquad \text{... 3.3.5}$$

$$x_{1,i+1} = x_{1,i} + ax_{3,i+1} \qquad \text{... 3.3.6}$$

$$x_{2,i+1} = x_{2,i} + ax_{1,i+1} \qquad \text{... 3.3.7}$$

Let $x_{2,i+1}$ be the variable which would be computed if there were no overflow, having a sign 's' where

$$s = +1 \text{ if } x_{2,i+1} > 0, \text{ and,} \qquad \text{... 3.3.8}$$

$$s = -1 \text{ if } x_{2,i+1} < 0 \qquad \text{... 3.3.9}$$

If the bound on the magnitudes of the variables is ±M, and this bound is exceeded during the addition of equation 3.2.22, then a two's complement overflow will give a variable $x_{2,ovf,i+1}$ where

$$x_{2,ovf,i+1} = x_{2,i+1} - 2sM \qquad \text{... 3.3.10}$$

Hence $$x_{2,ovf,i+1} - x_{2,i+1} = - 2sM \qquad \text{... 3.3.11}$$

The amount of overflow 'e' can be defined as a positive quantity using the relationship

$$x_{2,i+1} = sM + se \qquad \text{... 3.3.12}$$

$|x_{2,i}|$ is bounded by M, as is $|x_{1,i+1}|$. Hence, constraining coefficient a <= 1, the magnitude of the overflow

$$|e| < |ax_{1,i+1}| < M \quad \text{... 3.3.13}$$

Hence $$x_{2,ovf,i+1} + x_{2,i+1} = 2\, x_{2,i+1} - 2sM \quad \text{... 3.3.14}$$

$$= 2(sM + se) - 2sM \quad \text{... 3.3.15}$$

$$= 2se \quad \text{... 3.3.16}$$

The test function without an overflow is

$$Lf(x_{1,i+1}, x_{2,i+1}) = (1-ab)x_{1,i+1}^2 + x_{2,i+1}^2 - (a-b)x_{1,i+1}x_{2,i+1} \quad \text{... 3.3.17}$$

and with an overflow is

$$Lf_{ovf}(x_{1,i+1}, x_{2,ovf,i+1}) = (1-ab)x_{1,i+1}^2 + x_{2,ovf,i+1}^2 - (a-b)x_{1,i+1}x_{2,ovf,i+1} \quad \text{... 3.3.18}$$

Hence the between the magnitude of the test function when an overflow is present and that when the arithmetic operations are linear is

$$Lf_{ovf}(x_{1,i+1}, x_{2,ovf,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= [(1-ab)x_{1,i+1}^2 + x_{2,ovf,i+1}^2 - (a-b)x_{1,i+1}x_{2,ovf,i+1}] - [(1-ab)x_{1,i+1}^2 + x_{2,i+1}^2 - (a-b)x_{1,i+1}x_{2,i+1}]$$

$$\text{... 3.3.19}$$

$$Lf_{ovf}(x_{1,i+1}, x_{2,ovf,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= x_{2,ovf,i+1}^2 - x_{2,i+1}^2$$

$$- (a-b)x_{1,i+1}x_{2,ovf,i+1} + (a-b)x_{1,i+1}x_{2,i+1} \quad \ldots\ 3.3.20$$

$$= (x_{2,ovf,i+1} - x_{2,i+1})[(x_{2,ovf,i+1} + x_{2,i+1}) - (a-b)x_{1,i+1}]$$

$$\ldots\ 3.3.21$$

$$= (-2sM)(2se) + (a-b)x_{1,i+1}\ 2sM \quad \ldots\ 3.3.22$$

$$= 2sM[(a-b)x_{1,i+1} - 2se] \quad \ldots\ 3.3.23$$

since $\quad s^2 = 1 \quad \ldots\ 3.3.24$

The arithmetic operation giving the overflow is

$$x_{2,i+1} = x_{2,i} + ax_{1,i+1} \quad \ldots\ 3.3.25$$

Since the magnitude of $x_{2,i}$ is bounded by M then the sign of the overflow is the same as that of $x_{1,i+1}$.

$$x_{1,i+1} = s\ |x_{1,i+1}| \quad \ldots\ 3.3.26$$

Hence, since $s^2 = 1$,

$$Lf_{ovf}(x_{1,i+1}, x_{2,ovf,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= 2M[(a-b)|x_{1,i+1}| - 2e] \quad \ldots\ 3.3.27$$

which is positive if the overflow e

$$e < (a-b)|x_{1,i+1}|/2 \quad \ldots\ 3.3.28$$

Since it is possible for the test function to increase it is not a Liapunov function. Large signal limit cycles may exist. In fact such limit cycles do exist, and an example is given in figure 3.2. The time domain response, the test function and the corresponding phase portrait are given.

The locus on the phase portrait (x_1 as a function of x_2) of values for which the Liapunov function has a constant value is an ellipse. This ellipse is tilted from the x_1 and x_2 axes. The amount of tilt is related to the coefficients of the filter. The tilt means that a decrease in the magnitude of x_2 can give a point which is on a larger ellipse and hence a larger value of the test function.

3.3.2 Saturation Overflow in x_2

Liapunov analysis can be used to show that zero input (autonomous) limit cycles cannot arise because of a saturation overflow in the computation of x_2. Program steps can be included in the filter to give this saturation characteristic. The computation which gives the overflow in $x_{2,i+1}$ is

$$x_{2,i+1} = x_{2,i} + ax_{1,i+1} \qquad \ldots 3.3.29$$

Again let $x_{2,i+1}$ be the variable which would be computed if there were no overflow, having a sign 's' where

$$s = +1 \text{ if } x_{2,i+1} > 0, \text{ and,} \qquad \ldots 3.3.30$$

$$s = -1 \text{ if } x_{2,i+1} < 0 \qquad \ldots 3.3.31$$

If the bound on the magnitudes of the variables is ±M, and this bound is exceeded during the addition of equation 3.3.7, then saturation will give a variable $x_{2,ovf,i+1}$ where

$$x_{2,ovf,i+1} = sM \qquad \ldots 3.3.32$$

The test function without an overflow is

$$Lf(x_{1,i+1},x_{2,i+1}) = (1-ab)x_{1,i+1}^{2} + x_{2,i+1}^{2} - (a-b)x_{1,i+1}x_{2,i+1} \qquad \ldots 3.3.33$$

It has been shown that coefficients which give stable complex roots have bounds

$$0 < 1 - ab < 1, \quad \text{and} \qquad \ldots 3.3.34$$

$$|a + b| < 2 \qquad \ldots 3.3.35$$

It follows

$$|a - b| < 2 \qquad \ldots 3.3.36$$

The test function with an overflow is

$$Lf_{ovf}(x_{1,i+1},x_{2,ovf,i+1})$$
$$= (1-ab)x_{1,i+1}^{2} + x_{2,ovf,i+1}^{2} - (a-b)x_{1,i+1}x_{2,ovf,i+1} \qquad \ldots 3.3.37$$

Hence the effect of the overflow on the magnitude of the test function is

$$Lf_{ovf}(x_{1,i+1}, x_{2,ovf,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= [(1-ab)x_{1,i+1}^2 + x_{2,ovf,i+1}^2 - (a-b)x_{1,i+1}x_{2,ovf,i+1}] - [(1-ab)x_{1,i+1}^2 + x_{2,i+1}^2 - (a-b)x_{1,i+1}x_{2,i+1}] \quad \text{... 3.3.38}$$

$$= (x_{2,ovf,i+1} - x_{2,i+1})[(x_{2,ovf,i+1} + x_{2,i+1}) - (a-b)x_{1,i+1}] \quad \text{... 3.3.39}$$

$$= (sM - x_{2,i+1})(sM + x_{2,i+1} - (a-b)x_{1,i+1}) \quad \text{... 3.3.40}$$

$$= (M - sx_{2,i+1})(M + sx_{2,i+1} - s(a-b)x_{1,i+1}) \quad \text{... 3.3.41}$$

The first term of this product is always negative since the magnitude of the unsaturated value always exceeds that of the saturated value

$$sx_{2,i+1} > M \quad \text{... 3.3.42}$$

The second term is always positive since

$$M + sx_{2,i+1} > 2M \quad \text{... 3.3.43}$$

and, since $|x_{1,i+1}| < M$ and $|a-b| < 2$, then

$$|(a-b)x_{1,i+1}| < |(a-b)|M < 2M \quad \text{... 3.3.44}$$

Hence saturation overflow characteristic in the computation of x_2 cannot increase the test function. Therefore the function exhibits this Liapunov property and the second method of Liapunov can be used to guarantee that autonomous limit cycles in the digital filter section cannot arise because of saturation in the computation of x_2. The saturation characteristic is simple to program, and indeed is provided in some devices designed to implement digital filters. It remains to examine the behaviour of the filter with saturation during other computations, and to extend the analysis to inputs which are not zero.

3.3.3 Saturation Overflow in x_1

When applying Liapunov analysis to saturation in the computation of $x_{1,i+1}$ it is necessary to consider the sequential execution of the filter's program. The saturated value of $x_{1,i+1}$ is used to evaluate $x_{2,i+1}$, and this component will contribute to the value of the test function.

When saturation occurs during the computation of $x_{1,i+1}$ the program for the filter is

$$x_{0,i+1} = -bx_{1,i} \quad \text{... 3.3.45}$$

$$x_{3,i+1} = x_{0,i+1} - x_{2,i} \quad \text{... 3.3.46}$$

$$x_{1,ovf,i+1} = sM \quad \text{... 3.3.47}$$

$$= x_{1,i+1} - se \quad \text{... 3.3.48}$$

$$x_{2,ovf,i+1} = x_{2,i} + ax_{1,ovf,i+1} \quad \text{... 3.3.49}$$

$$= x_{2,i+1} - sae \quad \text{... 3.3.50}$$

The saturated value $x_{1,ovf,i+1}$ is set to the maximum magnitude, M. The sign s = +1 for a positive overflow and s = -1 for a negative overflow. Overflow has resulted in a quantity 'se' being subtracted from $x_{1,i+1}$ to give $x_{1,ovf,i+1}$. This definition ensures that the component e is positive. The overflow in $x_{1,i+1}$ has given a component 'sae' in $x_{2,ovf,i+1}$.

Hence $$x_{1,ovf,i+1} - x_{1,i+1} = - se \qquad \ldots\ 3.3.51$$

$$x_{1,ovf,i+1} + x_{1,i+1} = 2sM + se \qquad \ldots\ 3.3.52$$

and $$x_{2,ovf,i+1} - x_{2,i+1} = - sae \qquad \ldots\ 3.3.53$$

$$x_{2,i} + ax_{1,ovf,i+1} - x_{2,i+1} = -sae \qquad \ldots\ 3.3.54$$

$$x_{2,i+1} - a(x_{1,ovf,i+1}-se) = x_{2,i} \qquad \ldots\ 3.3.55$$

$$x_{2,i+1} - ax_{1,i+1} = x_{2,i} \qquad \ldots\ 3.3.56$$

The difference between the test function for the variables with saturation overflow with the corresponding function with linear addition is

$$Lf(x_{1,ovf,i+1},x_{2,ovf,i+1})-Lf(x_{1,i+1},x_{2,i+1})$$

$$= x_{1,ovf,i+1}^2 + (x_{2,ovf,i+1} - ax_{1,ovf,i+1})(x_{2,ovf,i+1} + bx_{1,ovf,i+1}) - x_{1,i+1}^2 + (x_{2,i+1} - ax_{1,i+1})(x_{2,i+1} + bx_{1,i+1}) \qquad \ldots\ 3.3.57$$

$$= (x_{1,ovf,i+1} - x_{1,i+1})(x_{1,ovf,i+1} + x_{1,i+1}) + x_{2,i}(x_{2,ovf,i+1} - x_{2,i+1} - b(x_{1,ovf,i+1} - x_{1,i+1})) \qquad \ldots\ 3.3.58$$

$$Lf(x_{1,ovf,i+1}, x_{2,ovf,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= \quad -se\ (2sM+se) \ - x_{2,i}(-sae \ + sbe) \qquad \ldots\ 3.3.59$$

$$= \quad -e\ [\ 2M+e + sx_{2,i}(a-b)] \qquad \ldots\ 3.3.60$$

This function is always negative because e and M are both positive, $|x_{2,i}|$ is bounded by M and, for coefficients of interest, $|a-b|$ is bounded by 2. There has been a reduction in the magnitude of the test function, and therefore it exhibits this Liapunov property. Since a Liapunov function exists, large signal saturation limit cycles cannot arise simply because of saturation overflows in the evaluation of x_1. The significance of the evaluation is that it has taken into account the effect of the nonlinearity on variables which have been computed later during the computation loop of the filter.

With some values of $x_{1,i}$ and $x_{2,i}$ there may be a saturation overflow in the evaluation of $x_{1,i+1}$, and also, in the same computation loop, a saturation overflow during the evaluation of $x_{2,i+1}$. It is necessary to show the Liapunov property applies in this case. Consider the evaluation of $x'_{2,ovf,i+1}$, the value of $x_{2,i+1}$ calculated using $x_{1,ovf,i+1}$ but with linear addition. The analysis of this section shows that the value $x_2'_{ovf,i+1}$ result in a reduction in the magnitude of the test function because the proof does not depend on $x'_{2,ovf,i+1}$ being bounded by $|M|$. Then the analysis for x_2 of the previous section applies directly to show that there will be a further reduction in the magnitude of the test function. Hence the test function exhibits this Liapunov property when saturation occurs when $x_{1,ovf,i+1}$ is evaluated and also when $x_{2,ovf,i+1}$ is evaluated. Liapunov analysis can be used to show that large signal limit cycles cannot occur because of saturations during the evaluations of variables x_1 and x_2. However there are three other points in this digital filter section where overflows can occur, and all these must examined in order to guarantee stability of the filter.

3.3.4 Saturation Overflow in x_0

Although the variables x_1 and x_2 are often chosen as the state variables of the proposed digital filter section, signal overflows may arise during any computation of the filter. An overflow may arise when the damping term is being computed when the damping coefficient $b > 1$. The error term introduced by this nonlinearity does alter successive variables during the same computation loop.

At the beginning of the $i+1^{th}$ computation loop the variables $x_{1,i}$ and $x_{2,i}$ are used to compute the next variables. If the operation of the program is linear its steps are

$$x_{0,i+1} = -bx_{1,i} \qquad \text{... 3.3.61}$$

$$x_{3,i+1} = x_{0,i+1} - x_{2,i} \qquad \text{... 3.3.62}$$

$$x_{1,i+1} = x_{1,i}+ax_{3,i+1} \qquad \text{... 3.3.63}$$

$$x_{2,i+1} = x_{2,i}+ax_{1,i+1} \qquad \text{... 3.3.64}$$

If a saturation overflow occurs during the computation of $x_{0,ovf,i+1}$ then

$$x_{0,ovf,i+1} = sM \qquad \text{... 3.3.65}$$

where s is the sign of the overflow.

It follows

$$x_{0,ovf,i+1} = -bx_{1,i} - se \qquad \text{... 3.3.66}$$

where 'e' is the magnitude of the error. The remaining program steps are

$$x_{3,ovf,i+1} = x_{0,ovf,i+1} - x_{2,i} \qquad \text{... 3.3.67}$$

$$= -bx_{1,i} - se - x_{2,i} \qquad \text{... 3.3.68}$$

$$x_{1,ovf,i+1} = x_{1,i} + ax_{3,ovf,i+1} \qquad \text{... 3.3.69}$$

$$= x_{1,i} + ax_{3,i+1} - ase \qquad \text{... 3.3.70}$$

$$= x_{1,i+1} - ase \qquad \text{... 3.3.71}$$

$$x_{2,ovf,i+1} = x_{2,i} + ax_{1,ovf,i+1} \qquad \text{... 3.3.72}$$

$$= x_{2,i+1} - a^2se \qquad \text{... 3.3.73}$$

The test function which is analyzed for Liapunov properties can be expressed as

$$Lf(x_1,x_2) = x_1^2 + (x_2-ax_1)(x_2+bx_1) \qquad \text{... 3.3.74}$$

The effect of the saturation overflow during the computation of $x_{0,ovf,i+1}$ on the magnitude of this function is

$$Lf(x_{1,i+1}, x_{2,i+1}) - Lf(x_{1,ovf,i+1}, x_{2,ovf,i+1})$$

$$= x_{1,i+1}^2 + (x_{2,i+1} - ax_{1,i+1})(x_{2,i+1} + bx_{1,i+1}) - [(x_{1,ovf,i+1})^2 + (x_{2,ovf,i+1} - ax_{1,ovf,i+1})(x_{2,ovf,i+1} + bx_{1,ovf,i+1})] \quad \ldots 3.3.75$$

This function can be rearranged to give

$$Lf(x_{1,i+1}, x_{2,i+1}) - Lf(x_{1,ovf,i+1}, x_{2,ovf,i+1})$$

$$= (2x_{1,ovf,i+1} + ase)ase + (x_{2,ovf,i+1} - ax_{1,ovf,i+1})(a(a+b)se) \quad \ldots 3.3.76$$

$$= ase(2x_{1,ovf,i+1} + (a+b)(x_{2,i}) + ase) \quad \ldots 3.3.77$$

$$= ase[(2x_{1,i} + asM) - abx_{1,i} - (a-b)(x_{2,i})] \quad \ldots 3.3.78$$

An examination of this expression shows that it can be negative, indicating that the saturation overflow in the computation of x_0 may increase the value of the test function.

Given that the overflow occurred during the computation

$$x_{0,i+1} = -bx_{1,i} \quad \ldots 3.3.79$$

the sign of $x_{1,i}$ is opposite to that of the overflow, 's', it follows that the component of the difference in the test functions,

$$ase(-abx_{1,i}) \quad \ldots 3.3.80$$

must be negative.

The term

$$ase[\ -(a-b)(x_{2,i})\ ] \qquad \ldots\ 3.3.81$$

may be negative, depending on the sign of $x_{2,i}$. The remaining term can be arbitrarily small. It follows the test function for the system with saturation can exceed the corresponding function for the linear system.

It is still possible that the Liapunov condition for the test function to decrease in magnitude at each computation can be met if the magnitude of the test function after the i+1th nonlinear computation is less than its value at the ith. This difference in the magnitudes is

$$Lf(x_{1,i},x_{2,i})-Lf(x_{1,ovf,i+1},x_{2,ovf,i+1})$$

$$= x_{1,i}^2 + (x_{2,i}-ax_{1,i})(x_{2,i}+bx_{1,i}) - [x_{1,ovf,i+1}^2+(x_{2,ovf,i+1}-ax_{1,ovf,i+1})(x_{2,ovf,i+1}+bx_{1,ovf,i+1})] \qquad \ldots\ 3.3.82$$

$$= x_{1,i}^2 + (x_{2,i}-ax_{1,i})(x_{2,i}+bx_{1,i}) - [(x_{1,i}-abx_{1,i}-ax_{2,i}-ase)^2 + (x_{2,i})(x_{2,i}+(a+b)x_{1,ovf,i+1})] \qquad \ldots\ 3.3.83$$

$Lf(x_{1,i},x_{2,i})-Lf(x_{1,ovf,i+1},x_{2,ovf,i+1})$

$$= a(2x_{1,i} - abx_{1,i} - ax_{2,i} - ase)(bx_{1,i} + x_{2,i} + se)$$
$$- (x_{2,i})[x_{2,i}+(a+b)(x_{1,i} + a(sM - x_{2,i})])$$
$$+ (x_{2,i}-ax_{1,i})(x_{2,i}+bx_{1,i}) \quad \ldots \; 3.3.84$$

$$= a/b[(sM-x_{2,i})sM(1-ab)-s^2e^2+x_{2,i}sM-b^2x_{2,i}(sM-x_{2,i})] \quad \ldots \; 3.3.85$$

Since the magnitude of $x_{2,i}$ is bounded by M, the first term

$$a/b(sM-x_{2,i})sM(1-ab) \quad \ldots \; 3.3.86$$

must be positive. However the magnitude of this term may be arbitrarily small because $x_{2,i}$ could approximate sM. Each of the remaining three terms may be negative, depending on the coefficient values and the value of $x_{2,i}$. Hence the magnitude of the test function may increase after saturation during the computation of $x_{0,i+1}$.

The magnitude of the test function for the system with a saturation nonlinearity may increase during the computation of $x_{0,i+1}$. Hence the corresponding Liapunov condition is not fulfilled. This does not guarantee that the filter will oscillate. There may be another function which does satisfy all the Liapunov conditions. Even if such a function cannot be found the filter may be stable.

It is possible to use additional program steps which prohibit autonomous large signal limit cycles. The overflow condition can be avoided. This can be achieved by limiting the magnitude of $x_{1,i}$ to M/b. Alternatively, when an overflow is detected, additional program steps can be used to compensate the magnitude of $x_{1,i+1}$ for the overflow.

The overall objective of material of this chapter is to consider the behaviour of the filter when its input is not zero. It is relevant that saturation overflows during the computations of the damping term can increase the magnitude of the test function, whereas the corresponding computations of variables x_1 and x_2 did not increase this magnitude.

3.3.5 Saturation Overflow in x_3

Again, if the operation of the filter is linear its program steps are

$$x_{0,i+1} = -bx_{1,i} \qquad \text{... 3.3.87}$$

$$x_{3,i+1} = x_{0,i+1} - x_{2,i} \qquad \text{... 3.3.88}$$

$$x_{1,i+1} = x_{1,i} + ax_{3,i+1} \qquad \text{... 3.3.89}$$

$$x_{2,i+1} = x_{2,i} + ax_{1,i+1} \qquad \text{... 3.3.90}$$

When a saturation overflow occurs during the computation of $x_{3,ovf,i+1}$ then

$$x_{0,ovf,i+1} = -bx_{1,i} \qquad \text{... 3.3.91}$$

$$= x_{0,i+1} \qquad \text{... 3.3.92}$$

$$x_{3,ovf,i+1} = sM \qquad \text{... 3.3.93}$$

where 's' is the sign of the overflow. It follows

$$x_{3,ovf,i+1} = x_{0,i+1} - x_{2,i} - se \qquad \text{... 3.3.94}$$

where 'e' is the magnitude of the error. Hence

$$x_{3,ovf,i+1} = -bx_{1,i} - se - x_{2,i} \qquad \text{... 3.3.95}$$

The remaining program steps are

$$x_{1,ovf,i+1} = x_{1,i} + ax_{3,ovf,i+1} \qquad \text{... 3.3.96}$$

$$= x_{1,i} + asM \qquad \text{... 3.3.97}$$

$$= x_{1,i+1} - ase \qquad \text{... 3.3.98}$$

$$x_{2,ovf,i+1} = x_{2,i} + ax_{1,ovf,i+1} \quad \text{... 3.3.99}$$

$$= x_{2,i} + a(1-ab)x_{1,i} - a^2x_{2,i} - a^2se \quad \text{.. 3.3.100}$$

$$= x_{2,i+1} - a^2se \quad \text{.. 3.3.101}$$

The effect of the error term 'e' on the magnitudes of $x_{2,ovf,i+1}$ and $x_{1,ovf,i+1}$ is the same as for an overflow in the computation of $x_{0,ovf,i+1}$. However the coefficient and the signal values which give the conditions for the overflow in $x_{3,ovf,i+1}$ do differ. Again the test function which is analyzed for Liapunov properties can be expressed as

$$Lf(x_1,x_2) = x_1^2 + (x_2-ax_1)(x_2+bx_1) \quad \text{.. 3.3.102}$$

The effect of the saturation overflow during the computation of $x_{3,ovf,i+1}$ on the magnitude of this function is

$$Lf(x_{1,i+1},x_{2,i+1}) - Lf(x_{1,ovf,i+1},x_{2,ovf,i+1})$$

$$= x_{1,i+1}^2 + (x_{2,i+1}-ax_{1,i+1})(x_{2,i+1}+bx_{1,i+1})$$

$$- [(x_{1,ovf,i+1})^2 + (x_{2,ovf,i+1}-ax_{1,ovf,i+1})$$

$$*(x_{2,ovf,i+1} + bx_{1,ovf,i+1})] \quad \text{.. 3.3.103}$$

Again this function can be rearranged to give

$$Lf(x_{1,i+1},x_{2,i+1}) - Lf(x_{1,ovf,i+1},x_{2,ovf,i+1})$$

$$= (2x_{1,ovf,i+1}+ase)ase + (x_{2,ovf,i+1}-ax_{1,ovf,i+1})(a(a+b)se)$$

$$\text{.. 3.3.104}$$

$$Lf(x_{1,i+1}, x_{2,i+1}) - Lf(x_{1,ovf,i+1}, x_{2,ovf,i+1})$$

$$= ase(2x_{1,i} + 2asM + 2ase + (a+b)(x_{2,i}) - ase) \quad .. \ 3.3.105$$

$$= ase/b(-2sM(1-ab) - (2-ab)se - (2-(a+b)b)x_{2,i}) \quad .. \ 3.3.106$$

If this difference were positive definite then the value of the test function with a saturation overflow would be less than the corresponding values if the computations were linear. However the expression has been arranged so that first two terms are negative, and last may be negative, depending on the sign of $x_{2,i}$.

It is possible that the Liapunov condition for the test function to decrease in magnitude at each computation can be met if the magnitude of the test function after the $i+1^{th}$ nonlinear computation is less than its value at the i^{th}. This difference in the magnitudes is

$$Lf(x_{1,i}, x_{2,i}) - Lf(x_{1,ovf,i+1}, x_{2,ovf,i+1})$$

$$= x_{1,i}^2 + (x_{2,i} - ax_{1,i})(x_{2,i} + bx_{1,i})$$
$$- [x_{1,ovf,i+1}^2 + (x_{2,ovf,i+1} - ax_{1,ovf,i+1})$$
$$*(x_{2,ovf,i+1} + bx_{1,ovf,i+1})] \quad .. \ 3.3.107$$

$$= x_{1,i}^2 + (x_{2,i} - ax_{1,i})(x_{2,i} + bx_{1,i})$$
$$- [((x_{1,i} - abx_{1,i} - ax_{2,i} - ase)^2$$
$$+ (x_{2,i})(x_{2,i} + (a+b)x_{1,ovf,i+1})] \quad .. \ 3.3.108$$

$$Lf(x_{1,i},x_{2,i}) - Lf(x_{1,ovf,i+1},x_{2,ovf,i+1})$$

$$= ase/b(-2sM(1-ab) - (2-ab)se - (2-(a+b)b)x_{2,i})$$
$$+ \ ab(x_{1,i}^2+(x_{2,i}-ax_{1,i})(x_{2,i}+bx_{1,i})) \quad .. \ 3.3.109$$

$$= a[b(x_{1,i}+se/b)^2 + se(sM(a+b)-se/b) + bsM(sM+(a+b)x_{1,i})] \quad .. \ 3.3.110$$

Although the first and the last two terms of this expression are positive definite these values approach zero as the filter's damping coefficient 'b' approaches zero. Under this condition the remaining term may be negative. Hence this arrangement of the difference

$$Lf(x_{1,i},x_{2,i})-Lf(x_{1,ovf,i+1},x_{2,ovf,i+1}) \quad .. \ 3.3.111$$

can be used to show that conditions exist where the value of the test function after the nonlinear computations may be greater than the corresponding magnitude before the computation loop. Hence this test function does not satisfy that Liapunov condition. Again the filter can be programmed to avoid autonomous large signal limit cycles using extra program steps when an overflow is detected to compensate the magnitude of $x_{1,i+1}$ for the overflow.

3.4 Application of Liapunov Analysis to Transients

The proofs of the previous sections show that a saturation overflow characteristic during the computation of x_1 or of x_2 cannot lead to autonomous limit cycles. The restriction of these proofs to autonomous limit cycles is very significant. The proofs do not apply if there is any non-zero input to the filter. Limit cycles may and do occur in these filters when non-zero inputs are present.

When determining the engineering requirement for the proof it is relevant that the filter would normally be designed so that its signal levels never exceed the bounds of linear operation. However if, by some means, a bound were exceeded, it is important the transient causing the overflow decays to zero so that the filter returns to its normal linear operation. Also the transient should decay at a suitable rate.

The following five variables can be associated with a saturation overflow during the computation of a variable, x.

Let x_{linear}, (x_{lin}) be the variable which would be computed if the arithmetic were linear.

$x_{bounded}$, (x_{bnd}) be the value which is present by the design which is such that each variable in the filter is operating within its operating range.

$x_{transient}$, (x_{trn}) be an additional component which is present in the computer which is not anticipated by the design. In general this transient is such that the bounds of linear operation are exceeded. (see equation 3.4.1).

$x_{saturated}$, (x_{sat}) be the value of the signal which is computed when saturation is present, and

$x_{difference}$, (x_{dif}) is the magnitude of the difference between the saturated value and x_{linear} which would be computed by the linear system.

It follows from these definitions that

$$x_{linear} = x_{bounded} + x_{transient} \qquad \ldots\ 3.4.1$$

$$= x_{saturated} + s x_{difference} \qquad \ldots\ 3.4.2$$

where s is the sign of the difference term (+1 or -1). Liapunov analysis is applied here to the transient component of the filter's response. It is possible to find conditions for the coefficients of the filter when the transient in the system with saturation nonlinearities will eventually decay to zero.

Liapunov analysis can be used to estimate the rate of decay of a transient component of a system. The magnitude of the Liapunov function gives bounds on the variables of the system. It has been shown that the magnitude of the Liapunov function for the digital filter section decreases in a constant proportion at each computation loop. This rate of decrease can be used as a bound on the magnitude of the transient variables to guarantee the rate of decay of the transient. For example if it can be shown that the arithmetic nonlinearity gives a further reduction to the Liapunov function, as is the case with a saturation overflow in x_1 or x_2, then Liapunov analysis can show that the rate of decay of the system variables is increased by the nonlinearity.

3.4.1 Transients with Saturation Overflow in x_2

The objective of this section is to consider the effect of the saturation overflow characteristic on the operation of a digital filter which has a non-zero input.

Consider the transient component in x_2 for saturation overflow condition. The five variables associated with an overflow in x_2 are

$$x_{2,lin} = x_{2,bnd} + x_{2,trn} \qquad \ldots\ 3.4.3$$

$$= x_{2,sat} + s x_{2,dif} \qquad \ldots\ 3.4.4$$

The test function considered for Liapunov properties is equation 3.2.21,

$$Lf(x_1,x_2) = x_1^2 + (x_2 - ax_1)(x_2 + bx_1) \qquad \dots 3.4.5$$

and in this section this function is applied to the variables associated with the transient. If the test function exhibits the Liapunov properties with respect to the transient component then the method of Liapunov can guarantee this transient component will decay to zero.

If the operation of the filter were linear the principle of superposition would hold giving

$$x_{0,trn,i+1} = - bx_{1,trn,i} \qquad \dots 3.4.6$$

$$x_{3,trn,i+1} = x_{0,trn,i+1} - x_{2,trn,i} \qquad \dots 3.4.7$$

$$x_{1,trn,i+1} = x_{1,trn,i} + ax_{3,trn,i+1} \qquad \dots 3.4.8$$

$$x_{2,trn,i+1} = x_{2,trn,i} + ax_{1,trn,i+1} \qquad \dots 3.4.9$$

In the remainder of section 3.4 the transient variables occur often and so the suffix 'trn' has been omitted. If an saturation overflow occurs during the computation of x_2 the computations of the variables give

$$x_{0,lin,i+1} = - bx_{1,lin,i} \qquad \dots 3.4.10$$

$$x_{3,lin,i+1} = x_{0,lin,i+1} - x_{2,lin,i} \qquad \dots 3.4.11$$

$$x_{1,lin,i+1} = x_{1,lin,i} + ax_{3,lin,i+1} \qquad \dots 3.4.12$$

$$x_{2,lin,i+1} = sM \qquad \dots 3.4.13$$

where s is the sign of the overflow and M is the maximum representable magnitude. The saturation nonlinearity has given a difference of magnitude $x_{2,dif,i+1}$ so that

$$x_{2,sat,i+1} = sM \qquad \ldots 3.4.14$$

$$= x_{2,lin,i} + ax_{1,lin,i+1} - sx_{2,dif,i+1} \qquad \ldots 3.4.15$$

The saturation overflow changes the magnitude of the test function. This change is

$$Lf(x_{1,i+1}, x_{2,i+1} - sx_{2,dif,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= (x_{2,i+1} - sx_{2,dif,i+1} - ax_{1,i+1})(x_{2,i+1} - sx_{2,dif,i+1} + bx_{1,i+1})$$
$$- (x_{2,i+1} - ax_{1,i+1})(x_{2,i+1} + bx_{1,i+1}) \qquad \ldots 3.4.16$$

$$= - sx_{2,dif,i+1}(x_{2,i+1} + bx_{1,i+1})$$
$$- (x_{2,i+1} - sx_{2,dif,i+1} - ax_{1,i+1})sx_{2,dif,i+1} \qquad \ldots 3.4.17$$

$$= - sx_{2,dif,i+1}(2x_{2,i+1} - (a-b)x_{1,i+1} - sx_{2,dif,i+1}) \qquad \ldots 3.4.18$$

If this difference were negative definite for all possible signal values and coefficients which give stable complex singularities it would have all the Liapunov properties and the transient would decay to zero. However the function is positive if

$$x_{2,dif,i+1} > 2sx_{2,i+1} - (a - b)sx_{1,i+1} \qquad \ldots 3.4.19$$

This condition can be satisfied with certain combinations of filter coefficients and signal values.

It is possible to clarify the implications of this inequality by comparing the result with that of section 3.3 where autonomous limit cycles were considered. The necessary condition for an increase in the magnitude of the test function was given by equation 3.4.19 as

$$x_{2,dif,i+1} - sx_{2,i+1} > sx_{2,i+1} - (a - b)sx_{1,i+1} \quad \ldots \; 3.4.20$$

When zero input limit cycles were considered the magnitude of the signal after saturation overflow was M.

Hence

$$x_{2,dif,i+1} - sx_{2,lin,i+1} > sx_{2,lin,i+1} - (a - b)sx_{1,lin,i+1} \quad \ldots \; 3.4.21$$

gives

$$-M > sx_{2,lin,i+1} - (a - b)sx_{1,lin,i+1} \quad \ldots \; 3.4.22$$

Since the magnitude of $sx_{2,lin,i+1}$ exceeds M for saturation, and the magnitude of $x_{1,lin,i+1}$ is bounded by M, and for the coefficients of interest $|a-b|$ is bounded by 2, this inequality cannot hold and therefore a decrease in the magnitude of the test function is guaranteed.

Returning to the consideration of the transient component the amount of the transient which is present after the signal overflow is

$$x_{2,dif,i+1} - sx_{2,i+1} > sx_{2,i+1} - (a-b)sx_{1,i+1} \quad \ldots\ 3.4.23$$

In the case of the transient component $x_{2,dif,i+1} - sx_{2,i+1}$ may be arbitrarily small, as may be the transient component $sx_{2,i+1}$. The expression

$$x_{2,dif,i+1} - sx_{2,i+1} > sx_{2,i+1} - (a - b)sx_{1,i+1} \quad \ldots\ 3.4.24$$

can be arranged as

$$x_{2,dif,i+1} - sx_{2,i+1} > sx_{2,i} + bsx_{1,i+1} \quad \ldots\ 3.4.25$$

For a saturation overflow

$$x_{2,dif,i+1} - sx_{2,i+1} < 0 \quad \ldots\ 3.4.26$$

and hence, using the previous inequality, either or both $x_{2,i}$ or $x_{1,i+1}$ must be negative. This means the signal must be such that the overflow in x_2 must be of opposite sign to that of the previous transient component in that variable, or in the opposite sign to that of the transient component in x_1, or both. These requirements would place restrictions on the bounded signals which would support the overflow condition.

The Liapunov property of a monotonic decrease in the magnitude of the test function can be fulfilled if its magnitude is less than the magnitude of the test function at the previous iteration. This difference is

$$Lf(x_{1,i+1}, x_{2,i+1} - sx_{2,dif,i+1}) - Lf(x_{1,i}, x_{2,i}) \qquad \ldots\ 3.4.27$$

It has been shown (equations 3.2.9 to 3.2.13) that the magnitude of the test function decrease in the proportion 1 - ab at each computation loop, so this difference can be expressed as

$$Lf(x_{1,i+1}, x_{2,i+1} - sx_{2,dif,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})/(1-ab) \qquad \ldots\ 3.4.28$$

Since for coefficients of interest 1-ab is positive the sign of the difference is the same as the sign of the expression

$$(1-ab)Lf(x_{1,i+1}, x_{2,i+1} - sx_{2,dif,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= Lf(x_{1,i+1}, x_{2,i+1} - sx_{2,dif,i+1}) - Lf(x_{1,i+1}, x_{2,i+1})$$
$$- ab\ Lf(x_{1,i+1}, x_{2,i+1} - sx_{2,dif,i+1}) \qquad \ldots\ 3.4.29$$

$$= - sx_{2,dif,i+1}(1-ab)(2x_{2,i+1} - (a - b)x_{1,i+1} - sx_{2,dif,i+1})$$
$$-ab\ (x_{1,i+1}^2 + (x_{2,i+1} - ax_{1,i+1})(x_{2,i+1} + bx_{1,i+1}))$$
$$\ldots\ 3.4.30$$

$$= sx_{2,dif,i+1}^2 - sx_{2,dif,i+1}(1-ab)(2x_{2,i+1} - (a-b)x_{1,i+1})$$
$$-ab\ (x_{1,i+1}^2 + (x_{2,i+1} - ax_{1,i+1})(x_{2,i+1} + bx_{1,i+1}))$$
$$-ab((1-ab)x_{1,i+1}^2 - (a-b)x_{1,i+1}sM + M^2) \qquad \ldots\ 3.4.31$$

If this test function, $Lf(x_1,x_2)$ were to decrease in magnitude at each computation loop then the quadratic in $sx_{2,dif,i+1}$ must be negative for all possible signal levels, for all possible overflows, for $0 < sx_{2,dif,i+1} < sx_{1,i+1}$, and for all filter coefficients giving stable complex singularities.

In order to test for this property the quadratic in $sx_{2,dif,i+1}$ has a negative definite constant term over the required range of filter coefficients. When there is no overflow, $sx_{2,dif,i+1} = 0$, the computations of the filter are linear and the decrease in the test function reduces to that of the linear case. There is the need to show there are no roots of this expression for $0 < sx_{2,dif,i+1} < sx_{2,i+1}$. If it can be shown that the quadratic of the form

$$sx_{2,dif,i+1}^2 + k_1\, sx_{2,dif,i+1} + k_2 \qquad \ldots\ 3.4.32$$

where k_1 and k_2 are constants, is negative at $sx_{2,dif,i+1} = 0$ and also at $sx_{2,dif,i+1} = x_{2,i}$, then there can be no roots of the quadratic between values 0 and $x_{2,i}$. The value of quadratic at $sx_{2,dif,i+1} = x_{2,i+1}$ is

$$-(1-ab)[abx_{1,i+1}^2 + x_{2,i+1}^2] + (a-b)x_{1,i+1}x_{2,i+1} \qquad \ldots\ 3.4.33$$

If this value were negative definite it would have shown that the test function decreased for all coefficients of interest. However this value can be positive, indicating the magnitude of the test function can increase over the coefficient range of interest. The function can be examined to give bounds on the coefficients where it is negative definite in $x_{1,i+1}$ and $x_{2,i+1}$.

The expression

$$- (1-ab)[abx_{1,i+1}^2 + x_{2,i+1}^2] + (a-b)x_{1,i+1}x_{2,i+1} \qquad \ldots\ 3.4.34$$

is negative definite for all values of x_1 and x_2 when

$$[(a-b)/2]^2 < (1-ab)^2/ab \qquad \ldots\ 3.4.35$$

It is interesting to examining the expression as the damping coefficient 'b' approaches 0. This gives an increment in the test function of

$$- x_{2,i+1}(x_{2,i+1} - ax_{1,i+1}) \qquad \ldots\ 3.4.36$$

This increment is positive only if $ax_{1,i+1}$ exceeds $x_{2,i+1}$. Once again for an increase in the magnitude of the test function the magnitude of the signal overflow must exceed the magnitude of the increment in its value during that current computation loop. Such an overflow requires a special bounded input.

Although it is not possible to prove that an overflow in the transient at x_2 will not lead to an increase in the transient component for all stable filters, the Liapunov approach has given bounds on the coefficients within which the test function does satisfy the Liapunov conditions.

One approach to achieving a proof is to examine other test functions for the Liapunov property of monotonically decreasing. However the increase occurs for very small values of the damping coefficient b, and it can be shown that as coefficient b approaches zero the only function of the form

$$x_1^2 + k_1 x_2^2 + k_2 x_1 x_2 \qquad \ldots\ 3.4.37$$

which exhibits the Liapunov property is the given test function. However this test function has been used to show that the increase in the transient component can only occur for special conditions. These conditions are that the damping coefficient is small, and that the magnitude of the overflow exceeds that of the transient component computed during the computation loop. It can be inferred that these situations are unlikely to occur. It remains to apply the analysis to the other nonlinear computations of the filter.

3.4.2 Transients with Saturation Overflow in x_1

Again the program without a signal overflow in $x_{1,i+1}$ is

$$x_{0,lin,i+1} = - bx_{1,lin,i} \quad \ldots 3.4.38$$

$$x_{3,lin,i+1} = x_{0,lin,i+1} - x_{2,lin,i} \quad \ldots 3.4.39$$

$$x_{1,lin,i+1} = x_{1,lin,i} + ax_{3,lin,i+1} \quad \ldots 3.4.40$$

$$x_{2,lin,i+1} = x_{2,lin,i} + ax_{1,lin,i+1} \quad \ldots 3.4.41$$

If when $x_{1,lin,i+1}$ is computed a saturation overflow with sign 's' occurs then

$$x_{1,sat,i+1} = sM \quad \ldots 3.4.42$$

$$= x_{1,lin,i} + ax_{3,lin,i+1} - sx_{1,dif,i} \quad \ldots 3.4.43$$

The computation $x_{2,lin,i+1} = x_{2,lin,i} + ax_{1,lin,i+1}$ ... 3.4.44

becomes $x_{2,sat,i+1} = x_{2,lin,i} + ax_{1,sat,i+1}$ ... 3.4.45

$$= x_{2,lin,i+1} - asx_{1,dif,i} \quad \ldots 3.4.46$$

The difference between the test functions for the linear and the nonlinear systems is

$$Lf(x_{1,i+1}-sx_{1,dif,i}, x_{2,i+1} - asx_{1,dif,i}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= (x_{1,i+1}-sx_{1,dif,i})^2 - x_{1,i+1}^2 + (x_{2,i+1}-asx_{1,dif,i}-ax_{1,i+1} + asx_{1,dif,i}) * (x_{2,i+1}-asx_{1,dif,i} + bx_{1,i+1} -bsx_{1,dif,i}) - (x_{2,i+1}-ax_{1,i+1})(x_{2,i+1} + bx_{1,i+1}) \quad \ldots 3.4.47$$

$$= -x_{1,dif,i}[(2sx_{1,i+1}-x_{1,dif,i}) + sx_{2,i} (a+b)] \quad \ldots 3.4.48$$

If this expression were negative then the test function satisfies this Liapunov condition and it would be possible to show that the transient in the nonlinear system must decay to zero. Since the term $-x_{1,dif,i}$ is negative the difference in the test functions is negative if the remaining factor is positive. This factor is

$$(2sx_{1,i+1} - x_{1,dif,i}) + sx_{2,i}(a+b) \quad \ldots\ 3.4.49$$

$$= -sx_{1,dif,i} + 2x_{1,i+1} + sx_{2,i}(a+b) \quad \ldots\ 3.4.50$$

Again there are certain signal levels and coefficient values where this expression is positive, hence the test function may not fulfill that Liapunov condition.

As with the overflow condition in x_2, again it is possible to compare the value of the test function after the nonlinear computations with its value at the completion of the previous computation loop.

$$Lf(x_{1,i+1} - sx_{1,dif,i}, x_{2,i+1} - asx_{1,dif,i}) - Lf(x_{1,i}, x_{2,i})$$

$$= Lf(x_{1,i+1} - sx_{1,dif,i}, x_{2,i+1} - asx_{1,dif,i}) - Lf(x_{1,i+1}, x_{2,i+1})/(1-ab) \quad \ldots\ 3.4.51$$

Since the term 1-ab is positive for coefficient values of interest, the sign of this function is given by

$$(1-ab)Lf(x_{1,i+1}-sx_{1,dif,i}, x_{2,i+1}-asx_{1,dif,i}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= Lf(x_{1,i+1}-sx_{1,dif,i}, x_{2,i+1}-asx_{1,dif,i}) - Lf(x_{1,i+1}, x_{2,i+1})$$
$$- ab\ Lf(x_{1,i+1}-sx_{1,dif,i}, x_{2,i+1}-asx_{1,dif,i}) \quad \ldots 3.4.52$$

$$(1-ab)Lf(x_{1,i+1}-sx_{1,dif,i}, x_{2,i+1}-asx_{1,dif,i}) - Lf(x_{1,i+1}, x_{2,i+1})$$

$$= (x_{1,i+1}-sx_{1,dif,i})^2$$
$$+ (x_{2,i+1} - asx_{1,dif,i} - ax_{1,i+1} + asx_{1,dif,i})$$
$$*(x_{2,i+1} - asx_{1,dif,i} + bx_{1,i+1}-bsx_{1,dif,i}))(1-ab)$$
$$- [x_{1,i+1}^2 + (x_{2,i+1} - ax_{1,i+1})(x_{2,i+1} + bx_{1,i+1})] \quad \ldots 3.4.53$$

$$= (1-ab)(sx_{1,dif,i})^2$$
$$- sx_{1,dif,i}(1-ab)(2x_{1,i+1} + (x_{2,i+1}-ax_{1,i+1})(a+b))$$
$$-ab(x_{1,i+1}^2 + (x_{2,i+1}-ax_{1,i+1})(x_{2,i+1}+bx_{1,i+1})) \quad \ldots 3.4.54$$

Again the objective is to show that this expression is not negative for the range of overflow conditions $0 < sx_{1,dif,i} < x_{1,i+1}$. At $sx_{1,dif,i} = 0$ the expression equals the original increment in the Lf so the expression is positive definite for the range of coefficients of interest. If it can be shown that at $sx_{1,dif,i} = x_{1,i+1}$ the expression is also positive definite then the Lf must decrease.

When $sx_{1,dif,i} = x_{1,i+1}$ then the increment in the test function is

$$-(x_{1,i+1}^2 + (x_{2,i+1}-ax_{1,i+1})(a+b)x_{1,i+1}+ab(x_{2,i+1}-ax_{1,i+1})^2) \qquad \ldots\ 3.4.55$$

$$= -[x_{1,i+1} + (x_{2,i+1}-ax_{1,i+1})(a+b)/2]^2 - (2ab-[(a+b)/2]^2)x_{2,i+1}-ax_{1,i+1}^2 \qquad \ldots\ 3.4.56$$

This expression must be negative when

$$[(a+b)/2]^2 > 2ab \qquad \ldots\ 3.4.57$$

For a positive increment in the test function (equation 3.4.56), signal values must be such that $x_{1,i+1} + (x_{2,i+1}-ax_{1,i+1})(a+b)/2$ is small. Though it may be possible to construct a bounded signal which gives a transient which increases in magnitude because of a saturation overflow condition, the previous equations show that it must fulfill several conditions in order to increase the magnitude of the test function.

i. The overflow must be a large proportion of transient signal.

ii. There is only a small range of coefficient values. These are very lightly damped.

iii. Since the test function decreases at each non-overflow computation where the operation of the filter is linear, situations with sustained growth in the transient would involve a large proportion of overflow in the computations.

Although Liapunov analysis has not shown that the magnitude of transients cannot increase in these filters when the arithmetic exhibits a saturation overflow characteristic, it has shown that situations giving a sustained increase are extremely rare situations. It remains to examine the corresponding properties of the remaining computations of the filter.

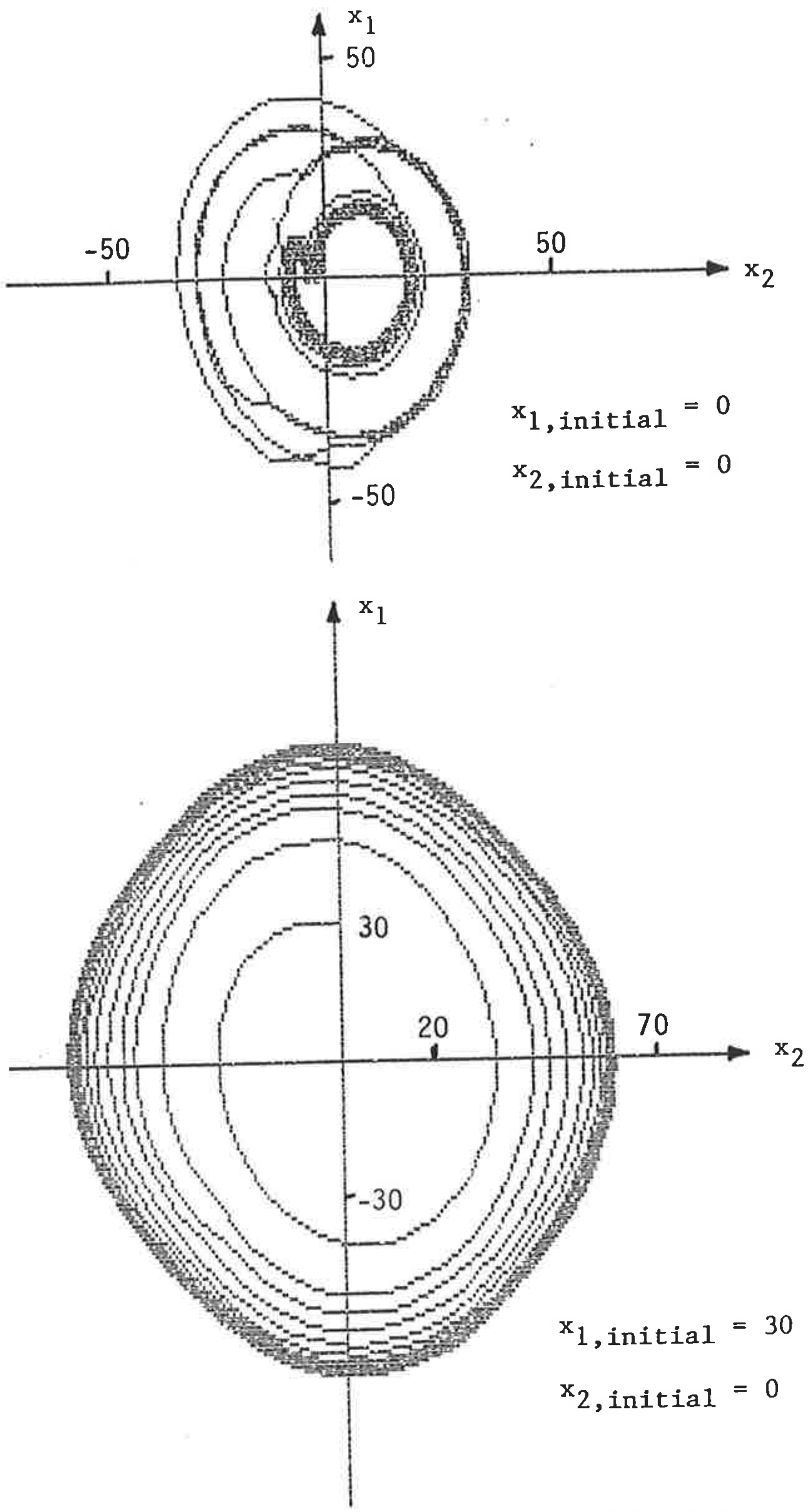
3.4.3 Transients with Saturation Overflow in x_0 and x_3.

The variables x_1 and x_2 are often defined as the state variables of the filter section, yet arithmetic overflows occur during the computations of all variables of the filter. When these occur during the computation of x_0 or of x_3 the resultant errors propagate to the remaining variables which are computed during the same computation loop of the filter section.

It has been shown (sections 3.3.4 and 3.3.5) that the function tested for Liapunov properties did not decrease monotonically when there was a saturation nonlinearity during the computation of x_0 or of x_3. This result was shown for the zero input condition, so the corresponding function of the transient variables may not decrease either. However the test function can be used to search for conditions where the transient may increase in magnitude. Then these conditions can be checked for their engineering significance.

Two identical filter sections are programmed, both with a saturation overflow in x_3. Filter 1 is started with zero initial conditions, and non-zero initial conditions are applied to the second filter. The second filter is used to search for an input which increases the magnitude of the test function. An input with +1 unit is applied, and an input of -1 unit, and the input which gives the greater magnitude of the test function of the transient variables selected.

The result of this search was to find an input which gives a transient which does not decay to zero. The significance is that the first filter, operating within its linear region has a transient applied to its initial conditions and this transient does not decay to zero. These results are given in figure 3.4.



a = 0.1, b = 0.15, saturation in x_3 at 40 units,
input magnitude, 10 units.

Figure 3.4 Phase Portraits of Differing Sustained Responses to the Same Input of Two Identical Filters. The Filters have Saturation During the Computation of x_3 and Different Initial Conditions.

Having used the Liapunov approach to find the input it remains to determine its engineering significance. A saturation overflow at x_3 gives an effective reduction in the gain around the loop of the filter. This reduction in gain then gives a reduction in the damped natural frequency of the poles of the filter section. For example a filter with a high Q which peaks at, say, 50 Hz may now give a response at 40 Hz. A 40 Hz input which will be attenuated by that filter when it is operating within its linear bounds. However a transient overflow resulting from differing initial conditions may lower the resonant frequency of the filter to 40 Hz. The result can be that a single transient will lead to a sustained overflow in a sinusoidal response in a filter. This is a situation of considerable engineering significance.

In order to avoid the overflows it is possible to program the filter so that the variables x_1 and x_2 are bounded, and then, using these bounds, programming so that overflows cannot occur during the computations of x_0 or of x_3.

3.5 Small Signal Limit Cycles

The finite registers of two's complement arithmetic give an underflow characteristic. The effect of a finite register length can be appreciated by considering a fictitious register of infinite length, and then setting to zero every bit which is less significant than the bit representing a quantization level, q. The result is to subtract a positive quantity from the variable which is not quantized until a quantization level is obtained. This quantity is called the quantization error. The quantization characteristic is that of the INT (integer) function of the BASIC computing language. The characteristic is given in figure 3.1.

A consequence of subtracting quantization errors is to reduce the magnitude of positive quantities, and to increase the magnitudes of negative. In certain cases there can be an increase in the gain around the loop and small signal limit cycles can occur. The intrinsic properties of the digital filter section considered here are that small signal limit cycles can arise, though by no means all the time. Figure 3.1 gives both a time domain representation of a small signal limit cycle, and also the corresponding phase portrait.

Two techniques for eliminating the limit cycles are presented here. The first is a guaranteed reduction in the magnitude of the Liapunov function. This is achieved by testing the magnitudes of the variables, x_1, and x_2, and then subtracting least significant bits so as to ensure a reduction in the Liapunov function. A guaranteed reduction in the Liapunov function is a sufficient but not a necessary condition for the elimination of limit cycles. It gives a severe reduction in the effective Q of the filter at low signal levels. For this reason a heuristic technique is introduced. Additional instructions are presented which have been shown to eliminate small signal limit cycles over many coefficient values and signal levels. However Liapunov analysis can not guarantee that stability of the system.

3.5.1 Guaranteed Reduction of the Liapunov Function

The filter section to be tested has the program sequence

$$x_{0,i+1} = - bx_{1,i} \qquad \dots 3.5.1$$

$$x_{3,i+1} = x_{0,i+1} - x_{2,i} \qquad \dots 3.5.2$$

$$x_{1,i+1} = x_{1,i} + ax_{3,i+1} \qquad \dots 3.5.3$$

$$x_{2,i+1} = x_{2,i} + ax_{1,i+1} \qquad \dots 3.5.4$$

The quantization characteristic depend on details of implementation of the filter. When each term is quantized the result is to add an error term which may be zero or negative. The multiplication

$$x_{0,i+1} = - bx_{1,i} \qquad \text{... 3.5.5}$$

can be achieved by the multiplication of coefficient 'b' by $x_{1,i}$. Quantization will add a non-positive quantity. Then the result must be negated in order to achieve the required transfer function, so with this implementation the effect of quantization has been to add a non-negative quantity. This is the implementation considered in every other section of the thesis.

In order to use Liapunov analysis to guarantee a reduction of the magnitude of the test function it is helpful to consider a different implementation of the program step

$$x_{0,i+1} = - bx_{1,i} \qquad \text{... 3.5.6}$$

It is possible to negate $x_{1,i}$ and then multiply this quantity by coefficient b. Since the last step in this implementation is the multiplication the effect of quantization is to add a non-positive quantity to the result.

There are two remaining places where quantization may occur in the filter. These result from the multiplications by 'a'. Each of multiplications results in the addition of a non-positive quantity to the result. Hence the program steps associated with the damping term have been chosen so that all quantization errors have the same sign.

Once the quantization effects are included the filter can be described using the quantization operator 'quant' which subtracts a quantization error, 'qe' where

$$0 <= qe < \text{quantization level} \qquad \ldots 3.5.7$$

The filter becomes

$$x_{0,i+1} = \text{quant}(-\, bx_{1,i}) \qquad \ldots 3.5.8$$

$$x_{3,i+1} = x_{0,i+1} - x_{2,i} \qquad \ldots 3.5.9$$

$$x_{1,i+1} = x_{1,i} + \text{quant}(ax_{3,i+1}) \qquad \ldots 3.5.10$$

$$x_{2,i+1} = x_{2,i} + \text{quant}(ax_{1,i+1}) \qquad \ldots 3.5.11$$

Hence

$$x_{0,i+1} = -\, bx_{1,i} \qquad - qe_1 \qquad \ldots 3.5.12$$

$$x_{3,i+1} = x_{0,i+1} - x_{2,i} \qquad - qe_1 \qquad \ldots 3.5.13$$

$$x_{1,i+1} = x_{1,i} + ax_{3,i+1} \qquad - qe_2 - aqe_1 \qquad \ldots 3.5.14$$

$$x_{2,i+1} = x_{2,i} + ax_{1,i+1} \qquad - qe_3 - aqe_2 - a^2qe_1 \qquad \ldots 3.5.15$$

The test function for this system can be expressed as the sum of two squares

$$Lf(x_1,x_2) = [x_2 - x_1(a-b)/2]^2 + x_1^2(1 - (a+b)^2/4) \quad \ldots 3.5.16$$

The magnitude of the test function will be reduced by the effects of quantization if quantization gives reductions in the magnitude of both the terms of equation 3.5.16. Since $(1 - (a+b)^2/4)$ is positive for coefficients of interest the second term is reduced by reducing the magnitude of x_1. For coefficient $a <= 1$ equation 3.5.14 can be used to show this reduction is achieved by adding up to two quantization levels whenever x_1 is positive. When coefficient a is between 1 and 2 it is necessary to subtract up to three quantization levels.

The reduction in the magnitude of the term, $x_2-x_1(a-b)/2$, is achieved by noting that for coefficients of interest

$$|(a-b)/2| < 1 \quad \ldots 3.5.17$$

and that, for coefficient $a < 1$, x_1 has quantization errors of magnitude up to 2 times the quantization level. To allow for this quantization error it is necessary to reduce the magnitude of x_2 by two quantization levels. Also the quantization in the computation to obtain x_2 gives a third quantization error, so, for coefficient a < 1, in order to guarantee a reduction in the magnitude of the second term it is necessary to add a third quantization level to negative values of x_2.

For coefficient a < 1 the program steps for the filter section become

x_0 = quant($-bx_1$) ... 3.5.18

$x_3 = x_0 - x_2$ ... 3.5.19

$x_1 = x_1$ + quant(ax_3) ... 3.5.20

$x_2 = x_2$ + quant(ax_1) ... 3.5.21

if $x_1 < 0$ then $x_1 = x_1 + 1$ ... 3.5.22

if $x_1 < 0$ then $x_1 = x_1 + 1$ ... 3.5.23

if $x_2 < 0$ then $x_2 = x_2 + 1$ ... 3.5.24

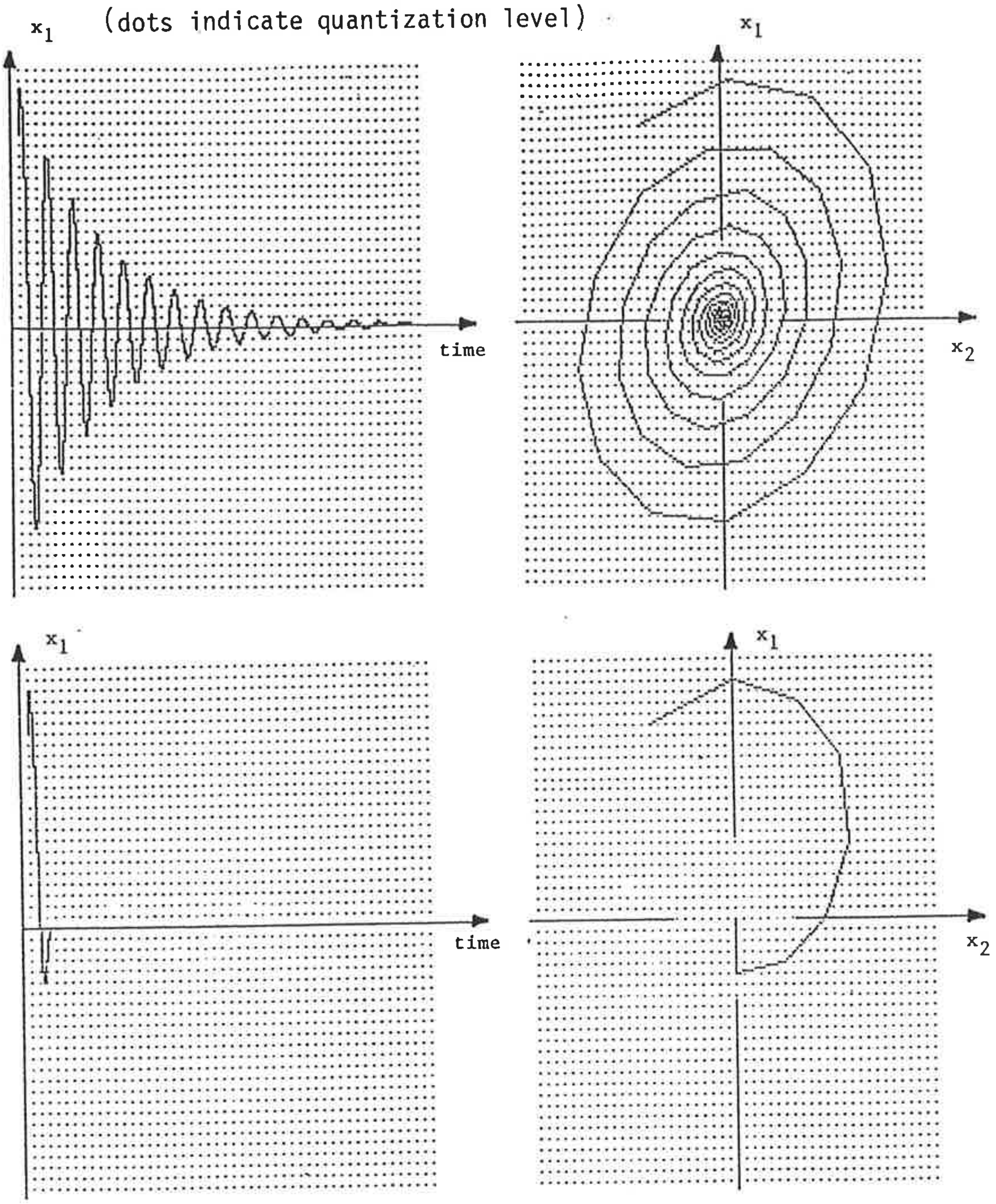
if $x_2 < 0$ then $x_2 = x_2 + 1$ ... 3.5.25

if $x_2 < 0$ then $x_2 = x_2 + 1$ ... 3.5.26

if $x_2 > 0$ then $x_2 = x_2 - 1$ ... 3.5.27

if $x_2 > 0$ then $x_2 = x_2 - 1$ ... 3.5.28

Figure 3.5 gives time domain representation and the phase portrait of the response of this filter. It has a relatively damped characteristic when compared with the filter having variables which are not quantized.



$x_{2,initial} = -10q, \quad x_{1,initial} = 18q, \quad a = 0.5, \quad b = 0.1$

Figure 3.5 Comparison of the Response of a Filter having Variables which are not Quantized with the Corresponding Response of the filter having Quantized Variables and Additional Program steps to Guarantee a Reduction in a Liapunov Function

3.5.2 Heuristic Technique

The program steps which guaranteed the elimination of small signal limit cycles by reducing the Liapunov function at each computation loop gave a considerable reduction of the quality factor, Q, of the second order filter section. A reduction of up to three quantization levels is used, and this gives a considerable bias towards damping. This decreases the effective register length of the filter by nearly 2 bits. A detail of implementation is that it is necessary to access the variable x_2, which is actually the difference between the output and corresponding input. For the purpose of quantization this value requires a comparison between the non-zero input and the output value.

It is possible to search for program steps which will prevent sustained small signal oscillations yet will not fulfill the Liapunov condition corresponding to the test function of equation 3.5.16. In order to avoid oscillations it is necessary to decrease the gain around the loop at small signal levels. One technique which has been tested is given in the program with the steps of 3.5.29 to 3.5.34. Note that the original program sequence for computing the damping term has been used (the negation follows the multiplication of x_1 by b). With this technique it is possible that the quantization errors may increase the Liapunov function, though such an increase has not been detected during many simulations.

The program steps for the computation loop is

$$x_0 = - \text{quant}(bx_1) \quad \ldots 3.5.29$$

$$x_3 = x_0 - x_2 \quad \ldots 3.5.30$$

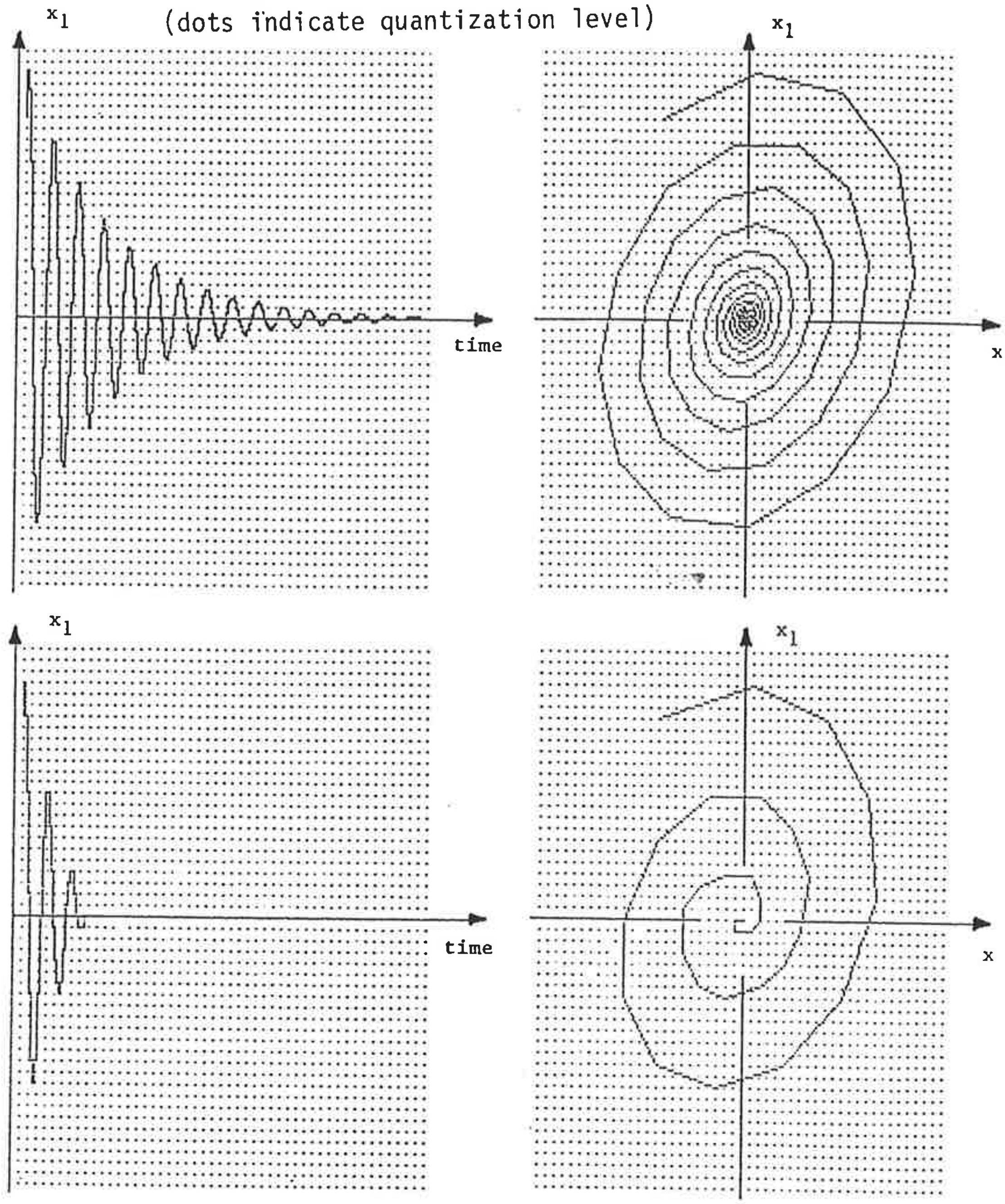
$$\text{if } ((x_0 > 0) \text{ and } (x_1 >= 0)) \text{ then } x_2 = x_2 + 1 \quad \ldots 3.5.31$$

$$\text{if } x_1 > 0 \text{ then } x_1 = x_1 - 1 \quad \ldots 3.5.32$$

$$x_1 = x_1 + \text{quant}(ax_3) \quad \ldots 3.5.33$$

$$x_2 = x_2 + \text{quant}(ax_1) \quad \ldots 3.5.34$$

Figure 3.6 gives the time domain representation and the corresponding phase portrait. Note that the effects of damping are less than the corresponding program for a guaranteed reduction in the magnitude of a Liapunov function.



$x_{2,initial} = -10q$, $x_{1,initial} = 18q$, $a = 0.5$, $b = 0.1$

Figure 3.6 Comparison of the response of a Filter with Variables which are not Quantized with that of the Corresponding Filter with Quantized Variables and Additional Program steps which have Eliminated Limit Cycles during Tests, but do not Guarantee a Reduction in the Magnitude of the Test Function.

3.6 Summary

The second method of Liapunov is a proof which gives a sufficient but not necessary condition for stability. It follows that when a Liapunov function is not found that the system may well be stable. Also, stabilizing techniques which give filters satisfying Liapunov's stability criterion may introduce considerable damping.

In the case of the saturation overflow characteristic, the method of Liapunov has been used to show that the results of overflows in the variables x_1 and x_2 may, in very rare situations cause oscillations. However overflow conditions can arise elsewhere in the filter, and the test function which is a Liapunov function for the linear filter, was used to find conditions where the behaviour of the saturating system departed from the linear in a way which is of engineering interest. The result is that in the design of a digital filter which must have a high reliability it is necessary to confine the nonlinearities to saturation overflows during the computations of x_1 and of x_2.

Although the quantization characteristics can lead to small signal oscillations, these are relatively rare even in the filter which has no special program steps to remove them. The Liapunov technique to guarantee the elimination of the limit cycles tends to stabilize the filter at low signal levels. Even so the Liapunov test function is a useful check for the operation of heuristic techniques to remove the limit cycles.

In consequence the designer of this filter section should

i. Place saturation bounds on variables x_1 and on x_2.

ii. Ensure that saturation cannot occur at x_0 or at x_3 using special program steps to account for the overflow, or by limiting the magnitudes of the input signal, of x_1 and of x_2.

iii. Use two conditional instructions (equns. 3.5.31 and 3.5.32) to remove the small signal limit cycles, which include, of course, any steady state errors.

These additional instructions can be used to ensure that the two's complement arithmetic will not cause sustained transients in the digital filter section.

Chapter 4 THE DESIGN OF CASCADE CONTROL COMPENSATORS

4.1 Introduction

There is a close relationship between digital filters and digital cascade compensators of control systems, (figure 4.1). Both devices modify the frequency components of signals. There is a fundamental difference between the applications which has implications for the design procedures. The digital filter is designed by considering its transfer function. By contrast the digital controller is in a system containing a given analog plant. [Bolton '81d]. In this application it is the response of the plant which is important. In this case the transfer function of the digital compensator is only a means to obtain the desired response.

The implications of this difference for the design procedures is that it is necessary to evaluate the response of the given plant in order to design the transfer function of the cascade compensator. Then the requirements of the compensator can be specified and the compensator designed.

One standard design procedure for sampled compensator design is to use the z domain. The transfer function of the given plant is transformed to the z domain, the system requirements can be

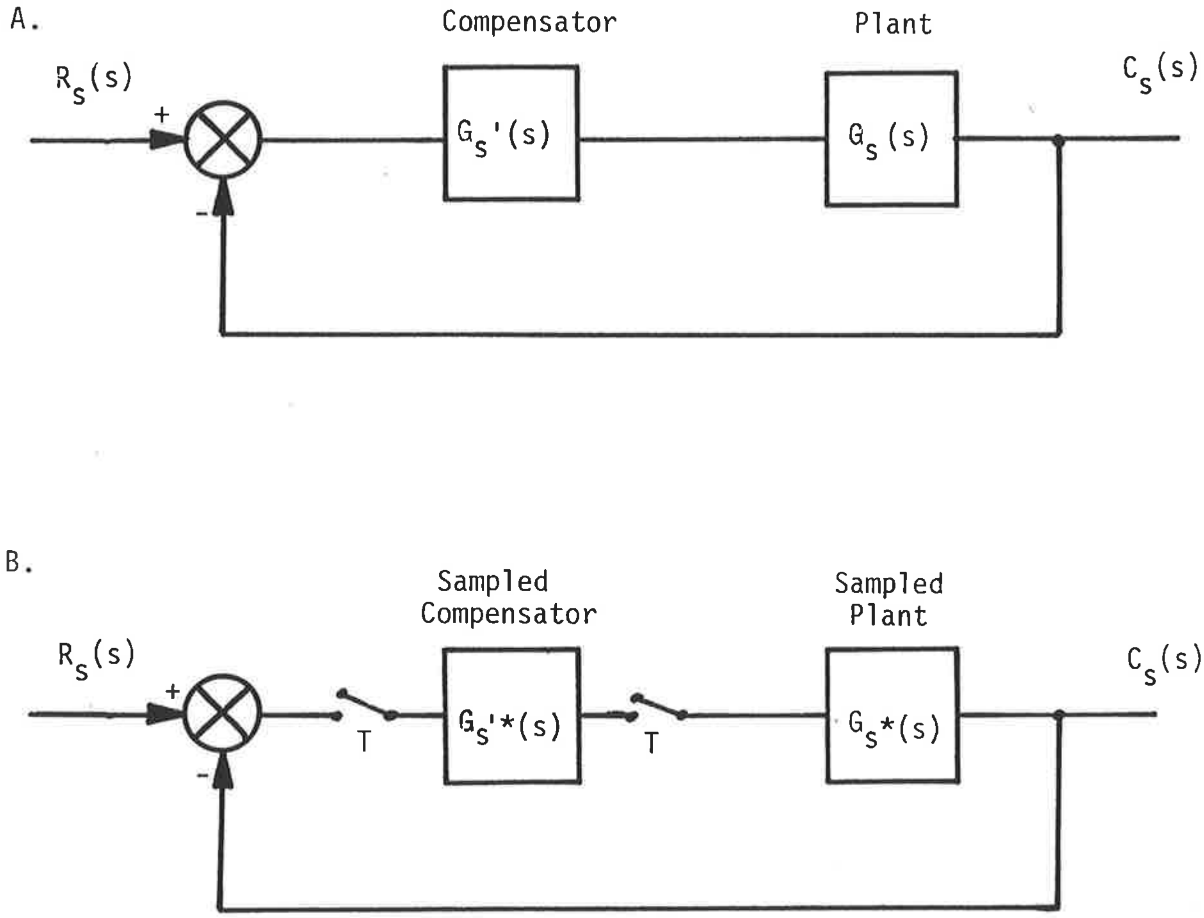


Figure 4.1 **Comparison of Sampled (B) and Continuous (A) Cascade Compensators.**

specified in the z domain and then the transfer function of the compensator determined. The approach used throughout this thesis is that in many applications sampling can be considered as introducing a slight change to a continuous system, and it is particularly useful to consider the equivalent continuous system.

The use of continuous domain techniques in the design of sampled compensators is not new. It is well known that the sinusoidal response of a sampled transfer function can be obtained as a frequency domain summation of the continuous transfer function. [Kuo '80]. This frequency domain summation can be interpreted as including the components which are introduced by sampling into the transfer function of the plant.

Similarly the conventional bilinear transform is often used to transform between the sampled and continuous domains. Usually a constant is introduced in order to transform the frequency response without approximation at a chosen frequency.

The frequency domain summation and the frequency warping compensated bilinear transform both use the frequency domain. Often control system design uses a point in the Laplace domain which has non-zero damping. This is a way of specifying the rise time and overshoot of the step response. The contribution here is to extend the conventional frequency domain techniques to the points in the Laplace domain which have non-zero damping. The contributions are

i. The summation for the transfer function of the sampled plant is extended to any point in the Laplace domain so that the plant's transfer function can be evaluated at a control system design point which has damping.

ii. Often the bilinear transform is compensated to allow for the effects of frequency warping. This gives equality between an continuous and a derived sampled transfer function at a selected point, again on the frequency locus. This procedure is extended to give equivalence between a given analog system and a derived digital system at any point in the Laplace domain. This equality is useful when designing sampled control systems.

iii. When implementing cascade control compensators it is useful to provide the results of the computation as quickly as possible. A technique is presented which reduces the delay through the compensator, though the total loop time is unchanged. The computations of the compensator's algorithm are carefully sequenced to achieve this result.

These contributions apply to a very few aspects of the overall control system design. Control system design involves the identification of the system, the choice of a suitable closed loop transfer function, and then the design of a compensator which will provide this system performance. Then there are further aspects to consider such as the sensitivity of the system to its parameters, the effects of non-linearities and bounds on the variables. In some situations an adaptive controller may be needed if the plant dynamics or system performance criteria vary considerably with time.

The design procedures are presented using the relatively simple classical situation where a single variable cascade compensator is to be designed for a linear system and the required location of the dominant closed loop pole pair is known. It is shown that a sampled compensator can be designed using Laplace domain techniques which correspond to the well known frequency domain procedures.

4.2 Review of the Sampled Closed Loop Design Procedures

Figure 4.1.A shows a continuous control system and figure 4.1.B gives the same system where sampling has been introduced at the compensator. Unity feedback has been used to simplify the example. The transfer function of the plant, $G_s(s)$, is given, and the compensator $G_s'(s)$ is to be designed. For the continuous system of figure 4.1.A the transfer function is

$$T_s(s) = \frac{C_s(s)}{R_s(s)} = \frac{G_s'(s)G_s(s)}{1 + G_s'(s)G_s(s)} \qquad \text{... 4.2.1}$$

and a design objective is to locate the dominant closed loop poles at a chosen value s_d. These poles are the roots of the characteristic equation

$$1 + G_s'(s)G_s(s) = 0 \qquad \text{... 4.2.2}$$

For the sampled system of figure 4.1.B, $G_s'^*(s)$ the sampled compensator must be designed. The symbol * denotes a sampled function. The transfer function of the sampled system is

$$T_s^*(s) = \frac{G_s'^*(s)G_s^*(s)}{1 + G_s'^*(s)G_s^*(s)} \qquad \text{... 4.2.3}$$

so the characteristic equation is

$$1 + G_s'^*(s)G_s^*(s) = 0 \qquad \text{... 4.2.4}$$

4.2.1 The Design Point $z_d = \exp(s_dT)$

The design procedure considered here involves choosing a point in the s plane s_d where the dominant closed loop poles of the compensated closed loop system are to be located. The compensator must be designed so that the characteristic equation is satisfied by s_d. Hence, for a continuous system

$$1 + G_s'(s_d)G_s(s_d) = 0 \qquad \text{... 4.2.5}$$

The corresponding procedure for the sampled system is to determine the design point in the z plane using the relationship $z_d = \exp(s_dT)$ which is the location of the impulse invariant z domain poles. The compensator must be designed so the characteristic equation is satisfied.

This equation is

$$1 + G_z'(z_d)G_z(z_d) = 0 \qquad \text{... 4.2.6}$$

The equivalent relationship in the Laplace domain using sampled functions is

$$1 + G_s'^{*}(s_d)G_s^{*}(s_d) = 0 \qquad \text{... 4.2.7}$$

The design of the cascade compensator includes three aspects.

i. Using the continuous plant transfer function $G_s(s)$ to find the sampled plant's transfer function at the design point $G_s^*(s_d)$. Section 4.3 presents a summation in the Laplace domain in order to evaluate the transfer function of the sampled plant.

ii. Obtaining the transfer function of the compensator which locates the dominant pole pair of the closed loop system at s_d. Section 4.4 contains a procedure for the design of a sampled compensator which uses an extension to the frequency warped bilinear transform.

iii. Finding a suitable algorithm (structure) for implementing the compensator. Section 4.4 contains a comparison of two structures which illustrates how delay through the cascade compensator may be minimized.

4.3 Evaluation of the Sampled Plant's Transfer Function

The objective of this section is to evaluate the sampled transfer function at the design point, $G_s^*(s_d)$, given the corresponding continuous transfer function, $G_s(s_d)$. The z domain procedure is presented for comparison with the proposed technique.

4.3.1 Z Domain Evaluation of $G_s'^*(s_d) = G_z'(z_d)$

The conventional z domain procedure for evaluating $G_z(z_d) = G_s^*(s_d)$ involves a partial fraction expansion of $G_s(s_d)$ in order to derive the z domain transfer function. In general the z transform of an all pole analog transfer function has zeros of transmission, and the design point s_d has been moved to $z_d = \exp[-sT]$. If a different sampling interval is to be considered the whole evaluation must be repeated. The procedure is illustrated for a representative plant, $G_s(s)$, which consists of two first order components

$$G_s(s) = \frac{0.3}{(s + 0.3)(s + 0.5)} \qquad \ldots \; 4.3.1$$

where the sampling interval T is normalized, (T = 1) with

$$s_d = -0.35 \pm j0.3. \qquad \ldots \; 4.3.2$$

Consider a compensator which incorporates a zero order hold at its output, as is usually the case (figure 4.2).

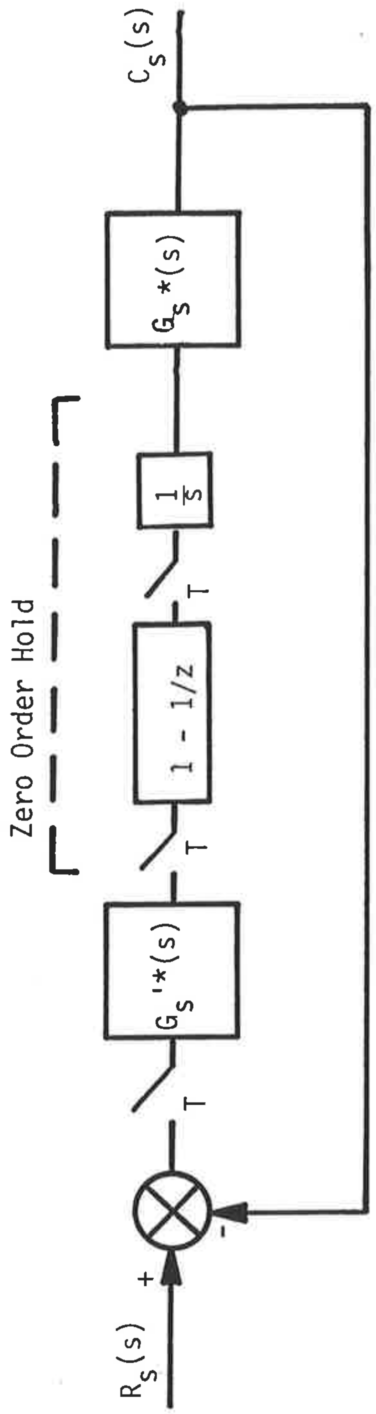


Figure 4.2 Zero Order hold at the Output of the Cascade Compensator.

The z domain evaluation uses the partial fraction expansion

$$\frac{G_s(s)}{s} = \frac{0.3}{s(s + 0.3)(s + 0.5)} \qquad \text{... 4.3.3}$$

$$= \frac{2}{s} - \frac{5}{s + 0.3} + \frac{3}{s + 0.5} \qquad \text{... 4.3.4}$$

The corresponding z transform of $G_s(s)/s$, $[G_s(s)/s]_z$, is

$$[G_s(s)/s]_z = 2z/(z-1) - 5z/(z-e^{-0.3}) + 3z/(z-e^{-0.5}) \qquad \text{... 4.3.5}$$

$$= 0.115501 \frac{z(z+0.765877)}{(z-1)(z-e^{-0.3})(z-e^{-0.5})} \qquad \text{... 4.3.6}$$

At the design point $s_d = -0.35 + j0.3$ so that for impulse invariance

$$z_d = e^{-0.35}\cos(0.3) + je^{-0.35}\sin(0.3) \qquad \text{... 4.3.7}$$

$$= 0.673214 + j0.20825 \qquad \text{... 4.3.8}$$

Using the characteristic equation 4.2.6, and incorporating the zero order hold, the sampled transfer function of the plant is

$$G_z(z_d) = \frac{0.115501 z_d(z_d+0.765877)}{(z_d-1)(z_d-e^{-0.3})(z_d-e^{-0.5})} \frac{z_d - 1}{z_d} \qquad \text{... 4.3.9}$$

$$= 3.5088 \text{ angle } [-172^{o}] \qquad \text{... 4.3.10}$$

The required compensator can be found using the characteristic equation. This gives

$$G_z'(z_d)[(z_d-1)/z_d] = -1/[G_s(s_d/s_d)] \quad \ldots 4.3.11$$

$$= 0.285067 \text{ angle } [-0.800468^{o}] \quad \ldots 4.3.12$$

$$\text{at } z_d = 0.673214 + j0.20825. \quad \ldots 4.3.13$$

4.3.2 Laplace Domain Summation of the Sampled Function

It is of interest that the continuous function $G_s(s_d)$ approximates the sampled function $G_s^*(s_d)$ in practical systems. A straight-forward substitution of $s = s_d$ in the equations for the continuous system (4.3.1 and 4.2.5) gives the required compensator

$$G_s'(s_d) = 0.284972 \text{ angle}[-8.00301^{o}] \quad \ldots 4.3.14$$

In this example sampling has introduced a change of less than one percent. In order to evaluate this difference using the z domain it has been necessary to perform a complete reformulation of the plant's transfer function from $G_s(s)$ to $G_z(z)$.

It is well known that the sampled function at any frequency, $G_s^*(j\phi)$ can be obtained as the sum of a series in the Laplace domain. The summation is [Kuo '80, (p401)]

$$G_s^*(j\phi) = 1/T \sum_{n=-N}^{n=N} G_s(j\phi + jn\phi_s) \qquad \ldots 4.3.15$$

where ϕ_s is the sampling frequency and N tends to infinity.

In order to extend this summation to any point in the Laplace domain it is possible to consider the sampling function as a harmonic series in ϕ_s. Then, for $s_d = r+j\phi$ an input $R_s(s_d)$ gives the sampled function $R_s^*(s_d)$. This is passed through the transfer function $G_s(s)$ to give $R_s^*(s_d)G_s(s_d)$, and finally at the output of the second sampler $[R_s^*(s_d)(G_s(s_d)]^*$ is obtained. In terms of double sideband modulation each harmonic term in the sampling waveform produces a term at $s_d \pm j\phi_s$ at the sampled output which is present at the input of the function $G_s(s)$. After passing through $G_s(s)$ function this component is sampled, again producing a component at s_d. In practical control systems the sampling frequency ϕ_s is greater than the magnitude of s_d and $G_s(s)$ has a low pass characteristic. In these cases the frequency translated terms are small when compared with the fundamental component.

The series for $G_s^*(s_d)$ is, [Bolton '81d].

$$G_s^*(s_d) = G_s(s_d) + \sum_{n=1}^{n=N} G_s(s_d+jn\phi_s) + G_s(s_d-jn\phi_s) \quad \ldots 4.3.16$$

In the numerical example of section 4.3.1 the series can be evaluated only to N = 1 giving

$$[G_s(s_d)/s]^* = -3.47495 - j0.488558 - 6.064\text{x}10^{-4} - j4.723\text{x}10^{-4} \quad \ldots 4.3.17$$

$$= -3.47555 - j4.89030 \quad \ldots 4.3.18$$

This summation is much simpler than using the z transform. The effect of sampling can be assessed by inspection, and a different sampling rate can be considered simply by choosing a different ϕ_s in the evaluation. Using this summation the required compensator has

$$G_s^*(s_d) = 0.284972 \text{ angle } [-8.0030^o] \quad \ldots 4.3.19$$

4.3.3 Accuracy of the Summation

When considering the contribution to the transfer function of the plant from the higher frequency terms it is possible to consider the transfer function as a net number of poles less a number of zeros. This assumption can be used to estimate the error resulting from truncating the series of equation 4.3.16. It follows the difference between the summation to N terms for

$G_s^*(s_d)$, $G_{s,N}^*(s_d)$, and its exact value is approximated by

$$G_{s,N}^*(s_d) = k \sum_{n=N+1}^{n=\infty} [1/(jn\phi_s)^{(P-Z)} + 1/(-jn\phi_s)^{(P-Z)}] \quad \ldots \; 4.3.20$$

$$\text{where} \quad G_s(s) = k \, \frac{s^P + \text{terms with lower powers of } s}{s^Z + \text{terms with lower powers of } s} \quad \ldots \; 4.3.21$$

These components add when difference between the number of poles in $G_s(s)$ and the number of zeros (P-Z) is even. In this case the error reduces by nearly a decade as each term in N is evaluated. This reduction is illustrated in figure 4.3. The error is very small for systems where P-Z is odd. This effect can be considered as arising because the cosine terms resulting from the sampling have become phase shifted to become sine terms and are nearly zero at the sampling instants. In practice it would not be wise to rely on this phase cancellation. Phase shifts at the relatively high sampling frequency would be sensitive to variations. In particular the higher order poles of the plant's transfer function which are usually neglected would have a considerable affect on the phase relationships.

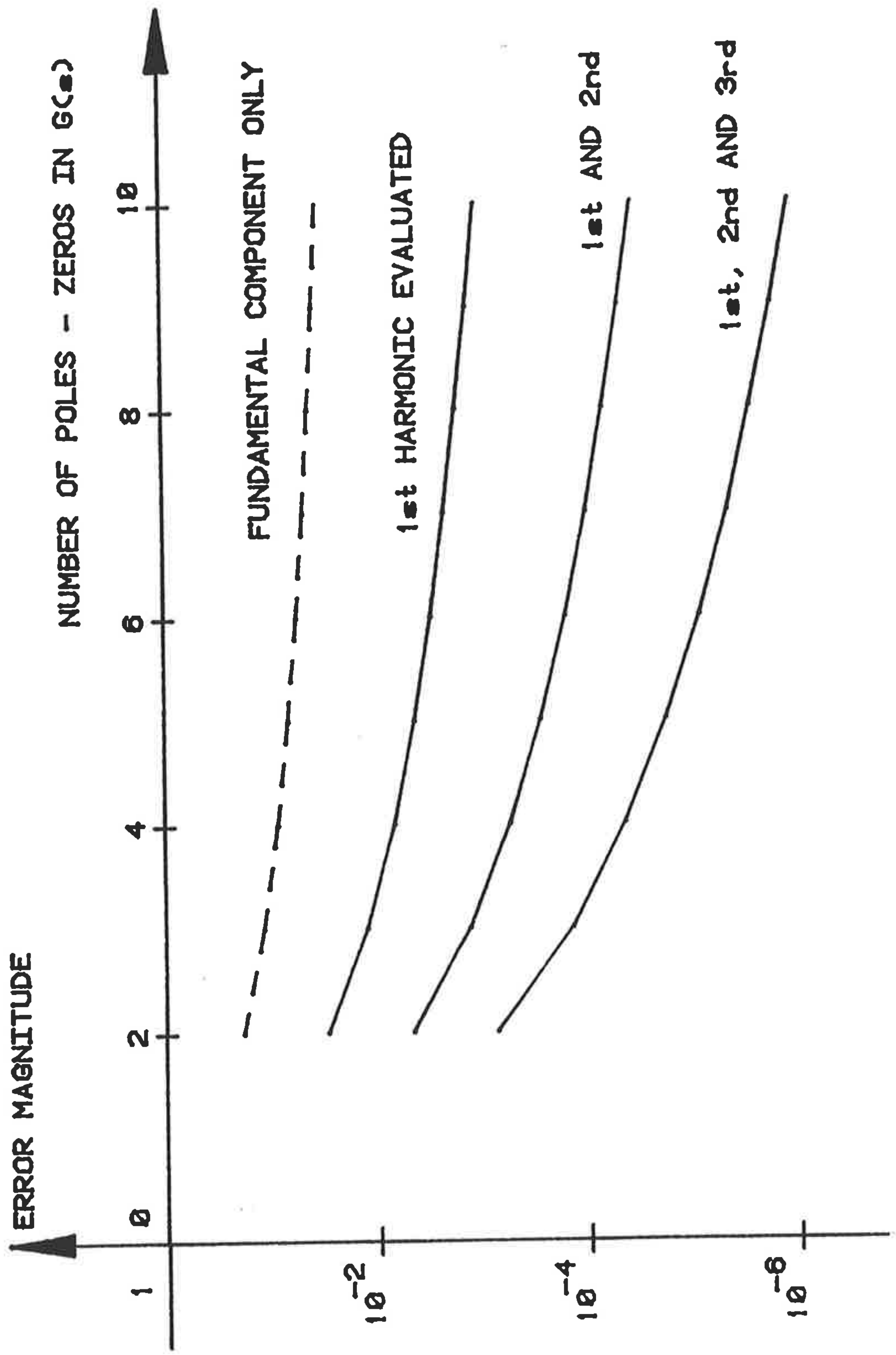


Figure 4.3 Maximum Error Magnitude in the Summation for $G_S^*(s_d)$.

In the example evaluated N is 1 and the difference between the number of poles and zeros, P-Z, is 3. Using equation 4.3.21 or figure 4.3 the bound on the error is 10^{-2}. In this particular case the net number of poles is odd which means the terms of equation 4.3.20 cancel. The actual error is less than 10^{-5}. In practical evaluations the series for $G_s^*(s_d)$ can be truncated whenever the magnitude of the added terms is sufficiently small. This is a well established procedure for evaluating the frequency domain summation, [Kuo '80] .

4.4 Damping Compensated Bilinear Transform

The bilinear transform between the Laplace and the z domain introduces warping of the frequency relationship. In digital filter design a factor, k_B, is often introduced to allow for this effect. The bilinear transform is

$$sT \Leftrightarrow k_B(z-1)/(z+1) = B(z) \qquad \ldots\ 4.4.1$$

where $$k_B = \phi_e \cot[\phi_e/2] \qquad \ldots\ 4.4.2$$

The factor k_B equates the original and the derived cutoff frequency of the filter at $s = j\phi_e$. A similar compensation can be made when the chosen value for s or z for equivalence involves damping as is the case with the design point of a sampled control system. This is the new transform presented in this section.

When digital filters are derived from an analog equivalent using the bilinear transform a sampled function $T_z(z)$ is derived from a continuous filter $T_s(s)$ where $T_s(s)$ suits the design specifications.

The approximation between $T_s(s)$ and $T_z(z)$ can be improved by increasing the order of the polynomials in z of the bilinear transform. This gives a corresponding increase in the order of the derived function $T_z(z)$ which makes $T_z(z)$ more difficult to implement. However transforms of the form

$$sT \Leftrightarrow k_B \frac{z - a_B}{z + b_B} = B(z) \qquad \ldots 4.4.3$$

where k_B ,a_B and b_B, are constants give $T_z(z)$ of the same order as that obtained using the bilinear transform of equation 4.4.1. The transforms of equation 4.4.3 do introduce z domain zeros in all pole analog functions as discussed in the context of digital filters. In general these require additional program steps for implementation. However with compensator design usually both pole(s) and zero(s) will be present in the Laplace domain and in the z domain transfer functions. In these cases altering the values of the constants from $a_B = 1 = b_B$ merely corresponds to adjusting the values of the coefficients which are already present in the compensator.

In the design of cascade control compensators it is useful to consider a point of equivalence $s_d = \sigma + j\phi$ where σ is non-zero. This is because equivalence in control systems is particularly important about the dominant closed loop poles which are damped.

4.4.1 Evaluation of the constants of the Bilinear Transform

The three constants k_B, a_B and b_B, can be determined by considering three constraints on the transform. These constraints are

i. the steady state gain of the compensator, because the steady state properties of the compensator are so important to the control system, and

ii,iii. the magnitude and phase of the equality at the design point $F[\exp(s_dT)] = \exp(s_dT)$.

The coefficient a_B is evaluated using equivalence in the steady state gain.

$$sT = B(\exp[sT]) \quad \text{at } s = 0 \qquad \ldots\ 4.4.4$$

$$s = 0 = B(0) = k_B(e^0 - a_B)/(e^0 + b_B) \qquad \ldots\ 4.4.5$$

Hence $a_B = 1$ ... 4.4.6

This is the value a_B of the conventional bilinear transform. With $a_B = 1$ the steady state error requirements are fulfilled without approximation by the derived digital compensator.

The coefficients k_B and b_B are evaluated using equality at the design point s_d

$$G_s'^*(s_d) = G_s'^*[B(\exp\{-s_d\})] = G_s'[\exp(-s_d)] \qquad \ldots\ 4.4.7$$

It follows

$$s_dT = B(\exp[s_dT]) \qquad \ldots\ 4.4.8$$

$$s_dT = \sigma_dT + j\phi_dT \qquad \ldots\ 4.4.9$$

$$= k_B(\exp[\sigma_dT+j\phi_dT] - 1)/(\exp[\sigma_dT+j\phi_dT] + b_B) \qquad \ldots\ 4.4.10$$

Hence:

$$(\sigma_d+j\phi_d)(\exp[\sigma_d][\cos(\phi_d)+j\sin(\phi_d)]+b_B)$$

$$= k_B(\exp[\sigma_d][\cos(\phi_d)+j\sin(\phi_d)]-1) \qquad \ldots\ 4.4.11$$

Considering the real and imaginary components of this equation, and solving for k_B and b_B.

$$k_B = \frac{\sin(\phi_dT)[(\phi_dT)^2+(\sigma_dT)^2]}{[\phi_dT\exp[-\sigma_dT]+\sigma_dT\sin(\phi_dT)-\phi_dT\cos(\phi_dT)]} \qquad \ldots\ 4.4.12$$

which reduces to $\phi_dT\cot(\phi_dT/2)$ when $\sigma_d = 0$ as with the conventional bilinear transform.

Also:

$$b_B = \frac{[\phi_d T\exp[\sigma_d T]-\sigma_d T\sin(\phi_d T)-\phi_d T\cos(\phi_d T)]}{[\phi_d T\exp[-\sigma_d T]+\sigma_d T\sin(\phi_d T)-\phi_d T\cos(\phi_d T)]} \quad \ldots 4.4.13$$

Equation 4.4.13 reduces to $b_B = 1$ when $\sigma_d = 0$, again giving the conventional bilinear transform.

At large sampling rates when T is small the expressions for k_B and b_B involves small differences between relatively large terms. For example the denominator contains the difference between the terms $\phi_d T\exp[-\sigma_d T]$ and $\phi_d T\cos(\phi_d T)$, both of which approach unity as T approaches zero. The small difference can be difficult to evaluate, particularly if the numeric representation within the computer has a few significant bits. In this case at large sampling rates the denominator can be evaluated using a series.

$$\text{denominator of equation 4.4.13} = \phi_d \sum_{i=2}^{\infty} (-1)^i \sigma_d{}^i / i!$$

$$+ \sigma_d \sum_{i=1}^{\infty} (-1)^i \phi_d{}^{(2i+1)}/(2i+1)!$$

$$- \phi_d \sum_{i=1}^{\infty} (-1)^i \phi_d{}^{2i}/(2i)! \quad \ldots 4.4.14$$

Similarly b_B can be evaluated using the summation

$$\text{summation} = 2\phi_d \sum_{i=1}^{\infty} (-1)^i \sigma_d^{2i+1}/(2i+1)!$$

$$- 2\sigma_d \sum_{i=1}^{\infty} (-1)^i \phi_d^{(2i+1)}/(2i+1)! \qquad \ldots 4.4.15$$

Then $$b_B = 1 - \frac{\text{summation}}{\text{denominator}} \qquad \ldots 4.4.16$$

4.4.2 Design Example

In the example considered the design point s_d is at -0.35+j0.3 and this value determines the constants b_B and k_B. Using equations 4.3.1 and 4.3.2 the damping compensated bilinear transform is

$$s = 1.75968 \frac{z - 1}{z + 0.48936494} \qquad \ldots 4.4.17$$

Again the sampling interval T has been normalized to unity. Substituting this expression for s in any analog compensator will preserve the steady state properties of the control system, as well as the location of singularities at s = -0.35+j0.3, by design the dominant closed loop poles.

If, for example, the design requirement was to increase the position constant (or steady state loop gain) of the control loop by a factor of 5 to reduce the steady state error, whilst locating the dominant closed loop poles at s_d, conventional analog control system design techniques give a suitable analog compensator as

$$G_s'(s) = \frac{5\ (11.108s + 1)}{169.43s + 1} \qquad \ldots\ 4.4.18$$

Using the damping compensated bilinear transform of equation 4.4.17, substituting into equation 4.4.18 gives

$$G_z'(z) = \frac{0.320321\ (z - 0.927791)}{z - 0.99504} \qquad \ldots\ 4.4.19$$

This equation is now in a form suitable for realization by a microprocessor algorithm. It remains to choose a structure, but when the transfer function is implemented the dominant closed loop poles which are solutions to the characteristic equation will be located at the design point as required.

4.5 Minimizing Delay through a Cascade Compensator

An important property of a cascade control compensator is the total time, T, taken to execute the algorithm which computes the forcing function. A second time interval is important also, δT

the delay between the sampling of the error signal and the availability of the corresponding forcing function. The final contribution is to present a sequence of instructions which minimize the delay through the cascade compensator, though not the total execution time. Two systems are compared in the program flowcharts of figure 4.4.

The delay δT is between the time at the input sample and the time the corresponding output is present at the forcing function of the plant. The delay for the structure of figure 4.4.A is much longer than that of 4.4.B. In the latter compensator all the instructions which do not involve the most recent data input value are computed before the current data value is read. Hence only a few remaining computations need be executed before the data is available at the output of the compensator.

The flowchart for the compensator with minimal delay can be derived by separating the terms in the transfer function which rely on the latest input value. For example the z domain transfer function of lag and of lead cascade compensators [Kuo,80] is

$$G_z'(z) = k_C \frac{z - a_C}{z - b_C} \qquad \ldots 4.5.1$$

Where the constants k_C, a_C and b_C which determine the transfer function of the compensator.

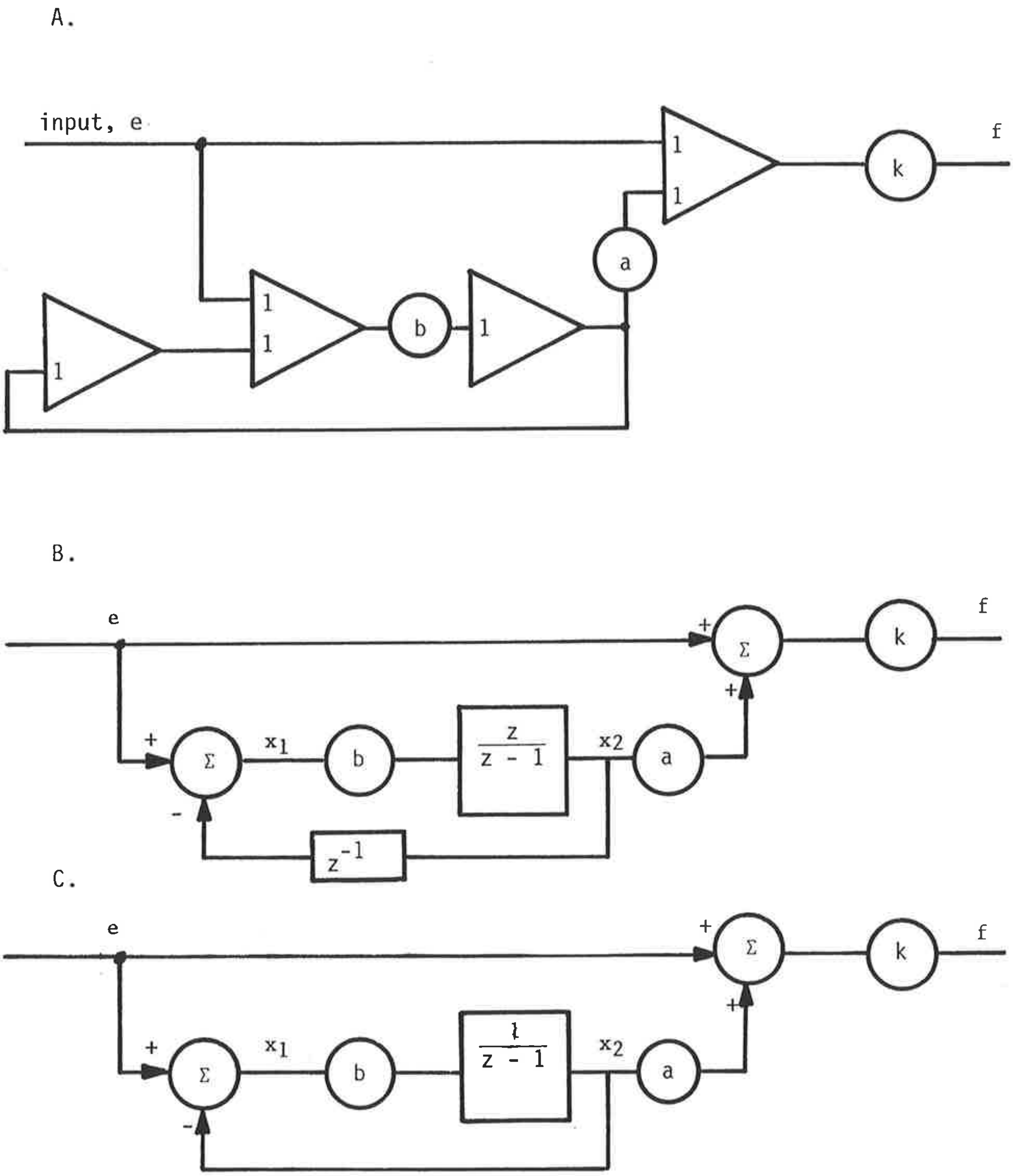


Figure 4.4 **Analog Computer Cascade Compensator with Two Digital Equivalents.**

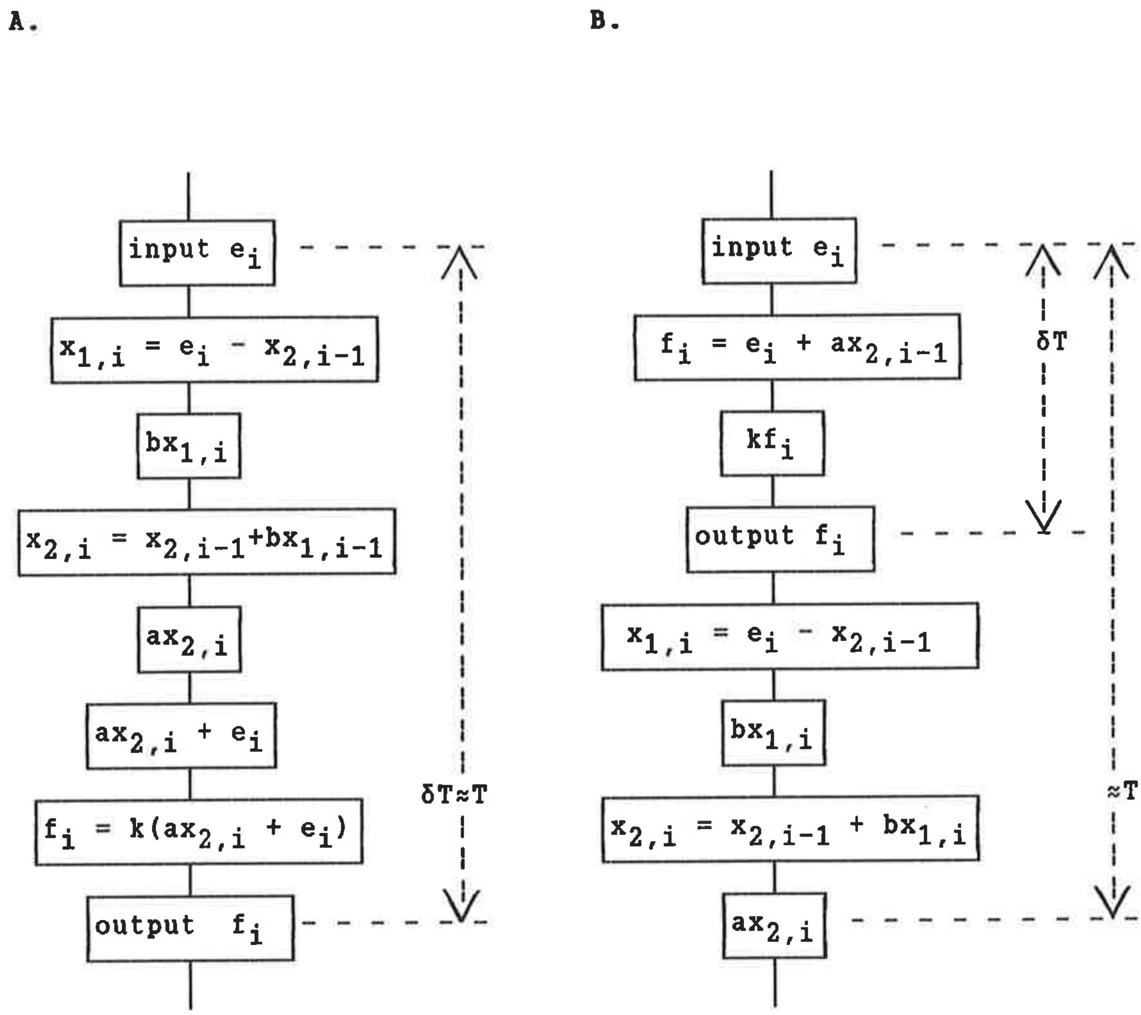


Figure 4.5 Flowcharts for the Block Diagrams of Figure 4.4.

(A) figure 4.4.B

(B) figure 4.4.C

Equation 4.5.1 can be rearranged to separate the computations which rely on the latest data value by separating the component in z^0. The transfer function becomes

$$G_z'(z) = k_C + k_C \, z^{-1} \, \frac{b_C - a_C}{z - b_C} \qquad \ldots \; 4.5.2$$

To minimize the delay δT, the microprocessor algorithm is simply designed to compute the expression with the delay before the current input data value is read.

In the program of figure 4.4.A the output f_i contains a component from the current input value e_i. By contrast the program of figure 4.4.B uses the previous integrated value. Small adjustments to the coefficient values can make the transfer functions of the systems identical, yet the lesser delay of the second compensator gives a better performance in a dynamic system.

4.6 Summary

In this chapter three new design techniques have been presented. The procedure for obtaining the sampled plant transfer function has been derived for a general point on the Laplace domain. Also the bilinear transform has been extended in a similar way. Then a procedure to minimize the delay through the compensator structure was outlined. It is routine to incorporate these design procedures in an computer design package in a way which is integrated with the conventional analog designs. The effects of sampling are clearly evident by inspection of the summation of equation 4.3.16, thus simplifying the difficult problem of selecting a sampling interval. Varying the sampling interval does not require a complete reformulation of the system.

Chapter 5 CONCLUSION

The objective of this thesis has been to use the relationships between continuous systems and sampled systems in order to devise some new design procedures for the sampled systems. This approach has been applied to digital filters and also to sampled cascade compensators of control systems. A digital filter is specified by the instructions whereas the control compensator has an additional element included which is the plant to be controlled. This difference gives corresponding differences to the design procedures. However there are many similarities.

A contribution of the thesis has been to extend a digital filter structure which is of digital incremental (DIC) form and has timing suited to program implementation. The extensions provide an infinitely deep notch output (with fewer additional instructions than available alternatives), and it has high pass and low pass responses available as internal signal variables. The structure has been extended to suit the signal magnitudes associated with high Q filter sections, [Bolton '84a].

Having proposed a versatile structure it was necessary to consider design techniques for the values of the filter coefficients. Design techniques have been devised to suit digital filters which have no feed forward terms (and are therefore all pole in the z^{-1} domain). These filters are of particular interest because of the reduction in execution time and program length. The thesis contains a new transform which is magnitude invariant between the continuous and sampled domains. The design procedure can be extended to compensate for the zero order hold at the output of the digital filter to a high degree of accuracy, [Bolton '84a]. Also the simple alternative of leaving the coefficients in the proposed digital filter structure unchanged from their analog equivalents has been shown to give very similar performance in the derived filter, even though the design procedure is extremely simple. Also, an all pole recursive Bessel digital filter has been developed for applications where the time domain (step response) properties of the digital filter are important, [Bolton '84d].

When implementing the digital filter it was necessary to examine the properties of the oscillations which arise when non-linear arithmetic is used. An aspect presented in this thesis is the consideration of the transient performance of the filters, rather than simply autonomous (zero input) limit cycles. It has been possible to apply Liapunov analysis to transient behaviour, and hence find examples where a saturation overflow characteristic gives a significant degradation to system performance. It has also been possible to specify program instructions which will avoid these effects for both large and small signal oscillations.

These new procedures give some alternatives to the experienced designer of digital filters. The conventional bilinear transform is suited to applications which require the implementation of zeros in the transmission. The new techniques can be used where a filter structure without feedforward terms can fulfill the design requirement and when it is not necessary to use general optimization design techniques.

The thesis contain analyses of the non-linear properties of the filter. These have been devised so the designer can be assured the response of the filter will not exhibit sustained transients. The non-linear behaviour of the filter has been examined for the case when non-zero inputs are present. Also all possible signal overflows throughout the filter have been considered.

Next it remained to examine the effect of including an analog element in the digital system which is then called a digital cascade compensator. Again digital systems have been designed by extending analog techniques. The thesis contains the damping compensated bilinear transform which extends the principles for the conventional bilinear transform to any point in the Laplace domain. This transform suits design procedures which leave the location of the dominant closed loop poles of the continuous and the sampled systems unchanged. Similarly it has been possible to use summations in the Laplace domain, an alternative to the conventional approach of using summations in the frequency domain, in order to provide alternative design procedures for control systems.

Similarly the designer of control system compensators now can use Laplace domain design techniques where frequency domain techniques are currently used, [Bolton '81d]. There has been a description of how to sequence the instructions in the compensator in such a way to minimize the time between the input to the compensator and the availability of the corresponding output (forcing function).

In summary, the objective of this thesis has been to provide new design procedures for digital filters which are simple to implement and which can be used in conjunction with the design procedures which are currently available.

REFERENCES

A.I.Abu-Et-Haija, K.Shenoi and A.M.Peterson. 'Digital Filter Structures Having Low Errors and Simple Hardware Implementation', IEEE Trans. Circuits and Syst., Vol. CAS-25, No.8, pp. 593-599, August 1978.

A.I.Abu-El-Haija and A.M.Peterson, 'A Structure for Digital Notch Filters', IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-27, No. 2, pp. 193-194, April 1979.

J.I.Acha and R.Robles-Diaz, 'Magnitude Response Compensation for Low-Pass Digital Filters', Int. J. Elect. Vol. 52, No. 5, pp. 471-476, 1982.

R.C.Agarwal and C.S.Burrus, 'New Recursive Digital Filter Structures Having Very Low Sensitivity and Roundoff Noise', Trans. Circuits and Syst., Vol. CAS-22, No. 12, pp. 921-927, Dec. 1975.

P.J.Allen, A.G.J.Holt, 'Implementation of a 2^{nd} Order Digital Filter Using AM2900 Bit Slice Devices and a Fast Multiplier', IEE Proc. Vol. 128, Pt. G, No. 4, pp. 216-223, Aug. 1981.

A.Antoniou and C.Shekher, 'Invariant-Sinusoid Approximation Method for Recursive Digital Filters', Elect. Lett., Vol. 9, No. 21, pp. 498-500, 18th Oct. 1973.

A.Antoniu, M.Degano, and C. Charalambous, 'Compensation for the Effects of the D/A Convertor in Recursive Digital Filters', IEE Proc., Vol. 129, Pt. G, No. 6, pp. 273-279, 1982.

K.J.Astrom and B.Wittenmark, 'Computer Controlled Systems', Prentice-Hall, ISBN 0-13-164319-3, 1984.

C.W.Barnes and A.T.Fam. "Minimum Norm Recursive Digital Filters which are Free of Overflow Limit Cycles", IEEE Trans. Circuits and Syst., Vol. CAS-37,, pp. 569-574, Oct. 1980.

R.E.Bogner and A.G.Constantinides Ed. 'Introduction to Digital Filtering', John Wiley, ISBN 0 471 08690 1, 1980.

A.G.Bolton, 'Y Transform, for the Design of Recursive Digital Filters', IEE Proc. G, Electron. Circuits and Syst., Vol. 127, No. 3, pp. 145-147, 1980,a.

A.G.Bolton, 'Coefficient Sensitivities in Recursive Digital Filters Implemented Using Microprocessors', IEE Proc., Vol. 127, Pt. G. No. 4, pp. 177-180, Aug. 1980,b.

A.G.Bolton, 'A Microprocessor Laboratory System to Process Analog Signals', Int. J. Elect. Enging. Educ., Vol. 17, pp. 233-242, Manchester U.P., 1980,c.

A.G.Bolton, 'Microprocessor Laboratory Procedure Within a Control Course', Int. J. Elect. Enging. Educ., Vol. 18, pp. 205-210, Manchester U.P., 1980,d.

A.G.Bolton, 'Representation and Simulation of Digital Filters', IEEE Circuits and Syst. Magazine, Vol. 2, No. 4, pp. 15-21, 1980,e.

A.G.Bolton. 'Microprocessor Laboratory Procedure to Introduce Digital Filtering', IEEE Trans. on Educ., Vol. E-24, No. 2, pp. 165-168, May 1981,a.

A.G.Bolton and B.R.Davis., 'Evaluation of Coefficient Sensitivities for 2nd Order Digital Resonators', IEE Proc., Vol 128, Pt. G, No. 3, pp. 127-130, June 1981,b.

A.G.Bolton, 'Design of Recursive Digital Filters for Microprocessor Implementation, Using Fixed Point Arithmetic', IEE Proc. G, Electron. Circuits and Syst., Vol. 127, No. 3, pp. 145-147, 1981,c.

A.G.Bolton, 'Design of Sampled Control Systems using Laplace Domain Techniques', IEE Proc. Pt. G, Electron. Circuits and Syst., Vol. 127, No 3, pp. 245-250, 1981,d.

A.G.Bolton, 'Transfer Functions Available with a New Second Order Digital Filter Structure', Proc. IEEE International Conference on Speech and Signal Processing, Atlanta, pp. 1217-1220, Mar. 1981,e.

A.G.Bolton, 'Filtering and Processing of Digital Signals', Chapter 10, Handbook of Fundamentals of Measurement Systems, Vol. 1, P.H.Sydenham Ed. John Wiley, 1982.

A.G.Bolton, 'An Analog to Digital Transform to Preserve the Frequency Magnitude Characteristics of All Pole Filters', IEE Proc. Pt. G, Electron. Circuits and Syst., Vol 131, No 3, pp. 131-133, 1984,a.

A.G.Bolton, "A Two's Complement Overflow Limit Cycle Free Digital Filter Structure", IEEE Trans on Circuits and Syst., Vol CAS-31, No. 12, pp 1045-1046, Dec. 1984,b.

A.G.Bolton, 'Zero Phase Filtering of Stored Data Using Recursive Digital Filters', IJE., vol. 56, No. 6, pp. 757-762, 1984,c.

A.G.Bolton, 'Bessel Digital Filters which are All Pole in the z^{-1} Domain', IJE., vol. 57, No. 5, pp. 761-765, 1984,d.

S.M.Bozic, 'Digital and Kalman Filtering', Edward Arnold, ISBN 0 7131 3410 0, 1979.

J.P.Brafman, J.Szczupak, and S.K.Mitra, 'An Approach to the Implementation of Digital Filters Using Microprocessors', IEEE Trans. on Acoust., Speech and Signal Processing, Vol. ASSP-26, No. 5, pp. 442-446, Oct. 1978.

M.U.A.Bromba and H.Zeigler, 'On Hilbert Space Design of Least-Weighted-Squares Digital Filters', Circuit Theory and Applications, Vol. 11, 7-32, pp. 8-32, 1983.

F.E.Burris, 'Design of Minimum-Phase Digital Filters from Linear Phase Prototypes', Elect. Lett., Vol. 10, No. 6, 21st Mar. 1974.

A.Buttle, A.G.Constantinides, J.E.Brignell, 'Online Digital Filtering', Electron. Lett., Vol. 4, No. 12, pp. 252-253, 14th June 1968.

C.Charalambous and A.Antoniou, 'Equalization of Recursive Digital Filters', IEE Proc., Vol. 127, Pt. G, No. 5, pp. 219-225, Oct. 1980.

A.T.Chottera and G.A.Jullien, 'A Linear Programming Approach to Recursive Digital Filter Design with Linear Phase', IEEE Trans. on Circuits and Syst. Vol. CAS-29, No. 3, pp. 139-149, Mar. 1982.

T.A.C.M.Claasen and L.O.G Kristiansson, 'Necessary and Sufficient Conditions for the Absence of Overflow Phenomena in a Second Order Digital Filter', IEEE Trans. on Acoustics, Speech and Signal Proc., Vol ASSP-23, No. 6, pp 509 - 515, Dec 1975.

A.G.Constantinides, 'Synthesis of Digital Chebyshev Digital Filters', Elect. Lett., Vol. 3, No. 3, pp. 124-126, Mar. 1967.

A.G.Constantinides, 'Elliptic Digital Filters', Elect. Lett., Vol. 3, No. 6, pp. 255-256, June 1967.

A.G.Constantinides, 'Frequency Transformations for Digital Filters', Elect. Lett., Vol. 3, No. 11, pp. 487-489, Nov. 1967.

A.G.Constantinides, 'Frequency Transformations For Digital Filters', Elect. Lett., Vol. 4, No. 7, pp. 125-116, 5th April 1968.

A.G.Constantinides, 'Digital Notch Filters', Elect. Lett., Vol. 5, No. 9, pp. 198-199, 1st May 1969.

A.G.Constantinides, 'Transition Ratio of Lowpass Digital Filters', Elect. Lett., Vol. 6, No. 4, pp. 87-89, 19th Feb. 1970.

A.G.Constantinides, 'Family of Equiripple Lowpass Digital Filters', Elect. Lett., Vol. 6, No. 11, pp. 351-353, 28th May 1970.

A.G.Deczky, 'General Expression for the Group Delay of Digital Filters', Elect. Lett., Vol. 5, No. 25, pp. 663-664, 11th Dec. 1969.

R.N.Gadenz and G.C.Temes, 'Computational Algorithm for the Design of Elliptic Filters', Elect. Lett., Vol. 8, No. 13, pp. 323-324, 29th June 1972.

R.Genesio and A.Laurentini, 'Tables for Butterworth-Digital-Filter Design', Elect. Lett., Vol. 6, No. 6, pp. 157-159, 19th Mar. 1970.

G.C.Goodwin and K.S.Sin, 'Adaptive Filtering Prediction and Control', Prentice-Hall, To be published.

C.J.Harris and S.A.Billings Ed, 'Self Tuning and Adaptive Control', IEE Control Engineering Series, Peter Peregrinus, ISBN 0 906048 62 1, 1981.

D.P.Henkel, D.J.Nowak and P.E.Schmid, 'Digital-Filter Precorrection for Zero-Order Hold Distortion', Elect. Lett., Vol. 4, No. 23, pp. 514-516, 15th Nov. 1968.

P.M.Hughes and B.M.G.Cheetham, 'The Generation of Equivalent Digital Filter Structures by a Modified Multiplier Extraction Approach', IEEE Trans. Circuits and Syst., Vol. CAS-29. No. 8, pp. 516-525, Aug. 1982.

K.J.Khatwani and A.K.Srivastava, 'On the Realization of Digital Filters Using a Variation of Partial Fraction Expansion', IEEE Trans on Circuits and Syst., Vol. CAS-25, No. 11, pp. 937-938, Nov. 1978.

N.G.Kingsbury, 'Second-Order Recursive Digital Filter Element for Poles Near the Unit Circle and the Real Z Axis', Electron. Lett. Vol. 8, No. 6, pp. 155-156. 23rd Mar. 1972.

N.G.Kingsbury, 'Digital-Filter 2nd order Element with Low Quantization Noise for poles and Zeros at Low Frequencies', Electron. Lett., Vol. 9, No. 12, pp. 271-273, 14th June 1973.

B.C.Kuo, 'Digital Control Systems' HRW, ISBN 0-03-057568-0, 1980.

J.Lefevre, 'Computer-Aided Design of All-Pass Digital Filters: Application to the Phase Linearization of Recursive Differentiators', IEE Proc., Vol. 129, Pt. G, No. 5, pp. 256, Oct, 1982.

P.A.Lynn, 'Economic Linear-Phase Recursive Digital Filters', Elect. Lett., Vol. 6, No. 5, pp. 143-145, 5th Mar. 1970.

H.C.Martinez and T.W.Parks, 'Design of Recursive Digital Filters with Optimum Magnitude and Attenuation Poles on the Unit Circle', IEEE Trans. on Acoustics Speech and Signal Proc., Vol. ASSP-26, No. 2, pp. 150-156, April 1978.

V.Mauro and A.R.Meo, Tables and Graphs for Designing Digital Chebyshev Filters', Elect. Lett., Vol. 6, No. 6, 19th Mar. 1970.

A.R.Memon, 'Determination of Pulse-Transfer Function of a Digital Filter from Prescribed Partial Requirements', Elect. Lett., Vol. 5, No. 19, pp. 465-466, 18th Sept. 1969.

A.R.Memon, 'Lowpass Digital Filters with Linear Phase', Elect. Lett., Vol. 6, No. 8, pp. 253-254, 16th Apr. 1970.

D.Mitra, 'Large Amplitude Self Sustained Oscillations in Difference Equations which Describe Digital Filter Sections Using Saturation Arithmetic', IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-25, No. 2, pp 134-143, April 1979.

H.T.Nagel Jr. and V.P.Nelson. 'Digital Filter Implementation on 16 Bit Microprocessors'. IEEE Micro., Vol. 1, No. 1, pp. 31. Feb 1981.

J.B.Naylor and J.M.Townsend, 'Lowpass Digital Filters Using Least-Squared-Error Design', Elect. Lett., Vol. 5, No. 25, pp. 647-649, 11th Dec. 1969.

G.Nielsen, 'Digital Processing of Analogue Signals: Part 1', Electronic Engineering, pp. 95-100, Nov. 1978.

G.Nielsen, 'Digital Processing of Analogue Signals: Part 2', Electronic Engineering, pp. 53-64, Dec. 1978.

G.Nielsen, 'Digital Processing of Analogue Signals: Part 3', Electronic Engineering, pp. 35-40, Jan. 1979.

A.V.Oppenheim and R.W.Schafer, 'Digital Signal Processing', Prentice-Hall, ISBN 0-13-214635-5, 1975.

C.L.Phillips and H.T.Nagle, 'Digital Control System Analysis and Design', Prentice-Hall, ISBN 0-13-212043-7, 1984.

L.R.Rabiner and B.Gold, 'Theory and Application of Digital Signal Processing', Prentice-Hall, ISBN 0-13-914101-4, 1975.

P.D.Roberts, 'Digital Filter with Independent Adjustable Parameters for use with a General Time-Weighted Mean-Square Error Criterion', Elect. Lett., Vol. 2. No. 4. pp. 150-151, Apr. 1966.

L.Shapiro, 'Information Theory in Digital Signal Processing', Electronic Engineering, pp. 37-56, June 1978.

L.Shapiro, 'The Design of Digital Filters: II', Electronic Engineering, pp. 35-63, Aug. 1978.

P.K.Sim and K.K Pang, ' Conditions for Overflow Stability of a Class of Complex Biquad Digital Filters, IEEE Trans Circuits and Syst., Vol CAS-34,No. 5, pp471 - 479, May 1987.

W.D.Stanley, 'Digital Signal Processing', Reston Publishing, ISBN 0-87090-199-1, 1975.

B.N.Suresh Babu and R. Yarlagadda, 'A Direct Approach to the Frequency Sampling Design of Digital Filters', Int. J. Electronics, Vol. 47, No. 2, pp. 123-138, 1979.

M.Terrell, H.Davenport and M.Ross, 'The Emerging World of Digital Filters', Electronic Engineering, pp. 26-86, Mid Oct. 1977.

T.J.Terrell, 'Introduction to Digital Filters', Macmillan, ISBN 0-333-24671-3, 1980.

P.Thajchayapong and P.J.W.Rayner, 'Design of Recursive Digital Filters Approximating to Arbitrary Prescribed Magnitude Responses', Elect. Lett., Vol. 10, No. 9, pp. 159-161, 2nd May, 1974.

Y.Tsypkin, 'Fundamentals of the Theory of Non-linear Pulse Control Systems', Proc. 2nd Int. Congr. Int. Fed. Automat. Contr., Basel Switzerland, pp. 172-180, 1963.

R.A.Valenzuela and A.G.Constandinides, 'Digital Signal Processing Schemes for Efficient Interpolation and Decimation', IEE Proc. Vol. 130, Pt. G, No. 6, pp. 225-235, Dec. 1983.

M.E.Van Valkenburg, 'Analog Filter Design', CBS College Publishing, ISBN 4-8338-0091-3, 1982.