# ISSUES IN THE DIGITAL IMPLEMENTATION OF CONTROL COMPENSATORS

by

PAUL MORONEY

S.B., Massachusetts Institute of Technology
(1974)

S.M., Massachusetts Institute of Technology
(1977)

E.E., Massachusetts Institute of Technology
(1977)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September, 1979

Signature of Author _____
Department of Electrical Engineering and Computer Science, September 1,1979

Certified by _____
Thesis Supervisor

Certified by _____
Thesis Supervisor

Accepted by _____
Chairman, Departmental Committee on Graduate Students

# ISSUES IN THE DIGITAL IMPLEMENTATION OF CONTROL COMPENSATORS

by

Paul Moroney

Submitted to the Department of Electrical Engineering and Computer Science

on September 1, 1979 in partial fulfillment of the requirements

for the Degree of Doctor of Philosophy.

## ABSTRACT

The implementation of control systems using small-scale digital hardware has largely been a neglected issue. However, in the field of digital signal processing a great deal of attention has been paid to the development of results concerning the finite-precision implementation of digital filters. In this thesis, we will *use*, *adapt*, and *extend* these ideas for digital feedback compensators. Specifically, we will primarily focus on steady-state linear-quadratic-Gaussian compensators. For some of the issues involved in compensator implementation, the filtering results apply directly; thus we can *use* existing concepts. However, in many cases, it will prove necessary to *adapt* these results. Finally, in our investigation we will uncover several extensions to the results as they apply to digital filters themselves. All three of these aspects are contributions to the development of digital control systems.

Name and Title of Thesis Supervisors:

Alan Steven Willsky
Associate Professor of Electrical Engineering

Paul Kenneth Houpt
Assistant Professor of Mechanical Engineering

## ACKNOWLEDGMENTS

No thesis is completed without the help of many other people. I would like to acknowledge the efforts of all these contributors, and offer them my sincerest thanks. In particular:

For the actual content of the thesis, I am deeply indebted to my supervisors Alan Willsky and Paul Houpt for their constant direction and encouragement, especially during the writing phase of the thesis. My enthusiasm and interest seemed perfectly matched to theirs. I would also like to thank my readers Jim McClellan, Gerry Prado, and Gunter Stein for their comments and help over the course of my research.

I also am indebted to the Charles Stark Draper Laboratory for their financial support and the resources that they have made available to me during the last two years.

In the preparation of the thesis, I was fortunate in being able to use the text editing system at MIT's Real-Time Systems group. Without the many willing hours put in by Clark Baker, John Moroney, and Jean Erickson, I never would have completed the final document on time. To them I owe a great thanks.

Finally, I would like to thank my parents for the moral support and encouragement that they have provided during my years at MIT — without their help I would not have made it.

## TABLE OF CONTENTS

## LIST OF FIGURES

## Chapter 1: Introduction

The design of time-invariant discrete-time compensators through the use of optimal regulators, pole-placement concepts, observer theory, optimal filtering [1,2,3], and also via classical control methods [4] has received a great deal of attention in the literature. In principle, these mathematical design procedures result in a compensator whose parameters are exact, that is, of infinite precision. In practice, such parameters are of double precision. Such a (near) ideal compensator has typically been implemented on large-scale floating-point computer systems, where high speed and accuracy are assured. Expense has not really been an issue. As a result, low-cost digital controllers have for the most part been quite simple, usually of the proportional-integral-derivative (PID) type [5].

The recent advances in digital hardware capabilities, such as the development of the microprocessor, have opened up many new applications for low-cost, real-time, small-scale digital control systems [5,6]. Thus the issues that arise in implementing compensators, that is, in approximating them with small-scale digital systems, cannot be ignored. Such issues include speed, finite memory limitations (finite precision), and expense. For its higher speed and lower cost, fixed-point arithmetic will be much preferred over floating-point (and assumed for this thesis). However, the effects of finite precision under fixed-point arithmetic are much worse than under floating-point. Such problems have not been addressed at all in the idealized mathematical design procedures that have been developed to date for control systems.

These idealized design procedures will result in an essentially infinite-precision transfer function for the compensator. The term *implementation* will refer to (1), the selection of a *structure* — the specification and ordering of the

computations that take place in the compensator during each sampling period, and also to (2), the selection of a hardware architecture and components. In implementing an ideal compensator, our aim is to produce a finite-precision digital system which either performs as close to the ideal as is consistent with the expense and speed requirements of the application, or which meets a specific level of performance relative to the ideal as inexpensively as possible subject to certain speed (sampling-rate) constraints. It is important to note that the mathematical design procedure which produces the ideal compensator and the implementation of this ideal compensator are not necessarily *independent* procedures; the initial design assumes a specific sampling rate, yet the implementation is frequently quite important in determining the maximum sampling rate.

Some effort has been directed towards investigating the issues involved in implementing digital feedback compensators, but it has been somewhat limited. Knowles and Edwards [7] and Curry [7.5] have each considered a roundoff noise analysis of certain sampled-data systems. Bertram [9], Slaughter [10], Johnson [11], and Lack [12] have developed amplitude bounds on the effects of quantization in sampled-data control systems. Sripad [13] has looked in some depth at the roundoff noise and finite-precision coefficient performance of the discrete-time Kalman filter and linear-quadratic-Gaussian controller. Rink and Chong [14] have derived bounds on the effects of quantization errors in floating-point regulators. Farrar [15] has pointed out in a basic way some of the issues involved in implementing continuous-time linear-quadratic-Gaussian controllers as discrete-time fixed-point microprocessor-based systems.

In his monograph, Willsky [16] has discussed a great number of parallels between the fields of digital signal processing and control and estimation. Many

of the basic issues involved in implementing digital feedback compensators have been examined in the context of digital signal processing, and a great many results exist. These digital filtering results are very important for control applications, since a digital control system can be viewed as a digital filter (the compensator) embedded in a feedback loop through a continuous-time plant. However, only in a few special cases do these results apply directly to control. Our task will be to *use*, *adapt*, and *extend* these results to the implementation of digital feedback compensators. In some cases we will directly use the filtering results. However, much of the time the control setting adds new twists to the implementation issues, requiring the adaptation of existing results. This effort, bridging two disciplines, is the most important contribution of the thesis. In addition, some of our work extends existing methods, or introduces new approaches, that are also useful for digital filtering applications. This contribution, although limited in scope, can be valuable to researchers in digital signal processing.

In this thesis, the steady-state linear-quadratic-Gaussian (LQG) control problem will be selected to convey our ideas on the implementation of feedback compensators. This type of controller has been shown to have desirable performance properties in terms of its robustness, multivariate formulation, optimal nature, and so forth. The LQG problem has also received a great deal of attention in the recent literature, and is being increasingly applied to real systems. Furthermore, the LQG problem has an explicit scalar objective function, which can be adopted as a performance metric against which the degradation due to finite wordlength effects can be measured. In fact, this was the degradation measure used by Sripad [13]. It is not necessary to choose this performance metric, or even use an LQG framework, but such a choice allows us to develop our results in

a concrete setting. Using this LQG control framework in the context of a single-input single-output system, we can bring out all the issues we wish to raise.

In Chapter 2 the details of the discrete-time LQG problem under consideration will be presented. Specifically, we will consider a continuous-time plant which is driven by additive white Gaussian noise and whose measured output is also corrupted by Gaussian noise and then sampled at rate $\frac{1}{T}$. The ideal discrete-time compensator will minimize an equivalent discrete-time performance index, subject to a piecewise-constant control signal $u(t)$. In presenting the equations for this ideal compensator, an important point will be raised. The finite calculation time implicit in the arithmetic operations of the compensator imposes a limit on the sampling rate of the system. Due to this same finite computation time, a realistic compensator must have its output at a given sample time depend only on *past* values of the compensator input. The *sample-skew* approach to this problem, which involves sampling the compensator input and output at different times [1], will also be presented in Chapter 2.

One of the important issues in discussing digital implementations is the notion of a *structure*. Given the system sample rate, the effects of finite precision on performance are dependent on the structure chosen, and not on the architecture or components selected. If all compensator computations can be performed with infinite precision, then all structures for implementing a given ideal compensator will be *equivalent* in performance. However, under the real constraint of finite precision, each structure will in general result in a different performance. Chapter 3 will describe some different compensator structures. Two important points will be stressed. First, the state space notation prevalent in control and estimation is not sufficient to represent all possible compensator structures. Second, the con-

cept of a structure for digital *filters* is not quite the same as the concept of a structure for digital *compensators*. This difference will require us to adapt the notation developed by Chan [17] for the representation of digital filter structures to the control case. A major implication of this change is that an $n^{th}$-order LQG compensator (for an $n^{th}$-order system) will have $n+1$ unit delay elements, and not $n$ as in the case of $n^{th}$-order filters. An important point will also be raised in Chapter 3 concerning the use of the ideal compensator equations resulting from the LQG design procedure *as a computational algorithm*; we can simply view this as one possible structure, what we will call the *simple* structure. We will show that, although this structure has been frequently used, more or less by default, it is not usually a good choice due to its large number of coefficient multiplications.

Architectural issues will be treated in Chapter 4. The ideas of *serialism* and *parallelism*, the degrees to which processes run sequentially or concurrently, will be presented in terms of the tradeoff they embody between compensator calculation time, which sets the maximum sample rate, and hardware complexity and expense. These ideas apply directly to digital compensators — no modifications are necessary. However the same cannot be said for the application of *pipelining* to control systems. The use of pipelining, a method for increasing the maximum sampling rate and performance of a system by altering the structure and the resulting transfer function in a very specific way in order to increase its inherent parallelism, raises an important issue — the interaction between the mathematical design of the ideal compensator and the finite-precision implementation of this ideal. The application of pipelining produces additional series delay in a compensator. If ignored, this delay will appear in a control system as extra negative phase shift, and perhaps cause instability. The only way to account for this de-

lay accurately will be to augment the discretized plant model and redesign the ideal compensator at the new sampling rate. Then if the same pipelining still applies to the new higher-order ideal compensator, *improved* performance can result.

In Chapters 5, 6, and 7 we will consider the effects of the finite memory limitations of inexpensive small-scale digital control systems. This restriction on memory will necessitate finite precision — the use of compensator coefficients (multipliers) of finite wordlength, and the insertion of quantization or overflow nonlinearities following the compensator input A/D converter and all multiplications (products) and additions. Methods must be found for selecting minimum coefficient and signal wordlengths which still result in acceptable levels of performance degradation, that is, in small-enough increases in the performance index.

Chapter 5 will treat the *uncorrelated* effects of product and A/D quantization on compensator performance. The major effort is spent on *roundoff* quantization, since the use of roundoff as opposed to *sign-magnitude* quantization results in lower levels of degradation, and also since roundoff effects can be analyzed in a tractable way. The main results of Chapter 5 are the adaptations of the scaling and roundoff noise analysis methods of digital filtering to the compensator case. There also arises an important implication concerning set-point LQG configurations and the scaling issue. Finally, minimum roundoff noise compensator structures will be adapted from the work of Mullis and Roberts [18] on minimum roundoff noise filter structures.

A sixth-order LQG control system will be introduced to test the roundoff analysis method of Chapter 5, and a number of different structures will be evaluated on the basis of their roundoff noise performance. We will show a significant similarity between the results for these structures and the results for

filter structures. However, two differences will arise. First, the potential presence of many real poles in a feedback compensator will complicate the pairing issue for parallel and cascade structures. Digital filters will typically have at most one real pole, so the pairing of such poles is of no interest. Second, although the default simple structure will perform relatively well, there will be two structures with many fewer coefficients that perform even better.

The effect of finite coefficient wordlength on performance is basically a deterministic one. Given any set of finite wordlength coefficients, we can compute exactly the resulting performance degradation, that is, the increase in the performance index. However, given a degradation level, it will be much harder to find the set of coefficients with the shortest wordlength that meets or exceeds this degradation level. If we make the common assumption that the ideal values of the coefficients will be *rounded* to finite wordlengths, then the wordlength determination can be accomplished with repeated evaluations of performance, one per wordlength tested. This procedure must also be repeated for each structure considered. Chapter 6 will describe the analytic methods developed for digital filters. Our emphasis will be on the use of a *statistical* measure of coefficient wordlength. For digital filters, this involves the use of first-order sensitivities with respect to the coefficients of the structure. However, for LQG compensators, all the first-order sensitivities will be zero, due to the optimal nature of the problem. Thus we will develop two new statistical estimates using *second-order* sensitivities. The necessity for second-order terms will exist for any parameter optimization problem, such as the sub-optimal reduced-order compensators described in Levine, Johnson, and Athans [19] and the sub-optimal decentralized controllers of Looze, Houpt, Sandell, and Athans [20]. In fact, if a digital filter is designed to

minimize some differentiable scalar function, then second-order sensitivities must be used for any statistical wordlength estimate based on that function. This will constitute an extension to the results for the implementation of digital filters.

We will test the same sixth-order control system and structures with the analytical procedures developed for coefficient wordlength effects. Again, we will show the similarity between our results and and the filtering results, and demonstrate that other structures with far fewer coefficients perform better than the simple structure. The statistical estimates of wordlength will be compared to the exact wordlengths required to meet a specific degradation level. We will show that the major advantage in using the statistical estimates is not in the computation time they may save over an iterative deterministic method, but in the fact that they are *continuous* and *differentiable* in nature. This fact allows us to apply iterative gradient minimization techniques to compute minimum coefficient wordlength structures, as described in Chapter 8. In this procedure, the bulk of the computations for the statistical estimates need be performed only once.

In Chapter 7, we will review the methods used in dealing with the *correlated* effects of the quantization and overflow nonlinearities present in a structure [21]. Any system including nonlinearities can exhibit oscillations, known as *limit cycles*. In digital filtering, there are three basic approaches to combatting such effects. First, we can use a structure that can be shown to have no limit cycles, given a specific type of nonlinearity. Second, the amplitude of any limit cycles can be upper bounded, allowing us to select a wordlength large enough to make this amplitude negligible. Finally, if a limit cycle occurs, we can inject enough roundoff noise to break up, or *quench*, the oscillation. Our results in this area for digital compensators are quite limited; however, several observations will be

made. First, a control system with an open-loop unstable plant or a plant with an integrator pole must of necessity have a low-amplitude limit cycle. Second, the global feedback loop around the compensator can alter the nature of any limit cycles that would occur in the open-loop compensator, and may even *cause* limit cycles. This point will be demonstrated for a finite impulse response compensator. (A finite impulse response *filter* is not recursive; therefore it can exhibit no limit cycles.) Finally, it is not clear that limit cycles will occur at all in LQG systems, given the system driving noise and measurement noise that is present. However, jump phenomena and other correlated noise effects may occur.

Chapter 8 will present a general iterative optimization technique for producing minimum roundoff noise and minimum coefficient wordlength structures. This procedure has been adapted from the optimization method of Chan for digital filters [17]. Essentially, this technique allows one to select a structure with a predetermined number of coefficients and iteratively vary those coefficients to minimize some scalar criterion. For LQG compensators, this criterion could be the increase in J due to roundoff noise or the increase due to finite wordlength coefficients, or some combination of these two. For the minimization of roundoff effects, the modification to Chan's procedure will be similar to the modification developed in Chapter 5 for roundoff analysis. However, the minimization of coefficient wordlength will require major changes since the statistical wordlength expression will actually be minimized, and this involves *second-order* sensitivities. The optimization procedure in Chapter 8 will also bring out two useful extensions for the *digital filtering* case. First, in minimizing roundoff noise effects, our procedure will be more general than that of Chan, accounting for the exact number of roundoff error sources and the location of each one in the structure. This gen-

eralization can be easily added to Chan's method. Second, we will set forth some general approaches to selecting which portion of a given structure to optimize, that is, the portion that will provide the greatest improvement when optimized. (An unconstrained optimization of the entire structure usually results in too many coefficient multipliers.) These guidelines also will apply to digital filter structural optimization.

Finally, Chapter 9 will review the contributions of this thesis, being careful to point out where our results are adaptations and applications of digital filtering techniques to the problem of implementing digital compensators, and where our results also constitute extensions to the digital filtering techniques.

## Chapter 2: The LQG Problem

A specific problem formulation is necessary to present in a unified manner the issues involved in implementing digital compensators. Historically, control theory has developed two different approaches — classical control (primarily a frequency-domain approach) and modern control (primarily a time-domain approach). For this thesis effort, we have selected the linear-quadratic-Gaussian (LQG) modern control problem for several reasons. The design of LQG systems has received a great deal of attention in recent times [3,22] due to its advantages for control (a multivariate nature, certain robustness properties [23], etcetera). As will be seen, the analysis of LQG compensators brings out all of the issues that we wish to discuss. Furthermore, the LQG problem has a very natural scalar objective criterion for determining its performance — the cost function $J$ (defined below). Such an objective function makes it quite simple to measure the degradation in performance resulting from any given compensator implementation. The most common criticism of the LQG approach, the difficulty in selecting the parameters of $J$ in some meaningful manner, is much less of a problem in light of the recent developments by Harvey and Stein [24] which relate frequency-domain design parameters to the selection of the scalar function $J$. This effort will thus help make the modern control approach more useful for small-scale low-cost digital systems. However, in principle the issues, approaches, and results developed here apply to any control and/or estimation implementation. This chapter will thus present the set of assumptions inherent in the LQG control problem and describe its discrete-time solution.

Consider a continuous-time plant whose performance is to be improved through feedback. Assume that the $n^{th}$-order state space equations (2.1) and

(2.2) accurately model the input-output behavior of the plant, including any sensor and actuator dynamics: (Brackets will indicate continuous-time quantities, while parentheses will indicate discrete-time quantities.)

$$\dot{x}[t] = A\,x[t] + B\,u[t] + w_1[t] \tag{2.1}$$

$$y[t] = C\,x[t] + w_2[t] \tag{2.2}$$

where the time-invariant system matrix $A$ is $n \times n$, the input gain matrix $B$ is $n \times m$,and the output gain vector $C$ is $p \times n$. The $n$-vector $x[t]$, $m$-vector $u[t]$, and $p$-vector $y[t]$ represent the system states, inputs, and outputs respectively. The $n$-vector $w_1(t)$ and $p$-vector $w_2(t)$ represent uncorrelated white Gaussian noise sources of covariances $\Xi_1$, and $\Xi_2$, where $\Xi_2 > 0$. It is further assumed that the performance of the system can be expressed as a scalar quantity which is a quadratic function of the states and controls:

$$J_c = E\left\{\lim_{\tau\to\infty} \frac{1}{2\tau} \int_{-\tau}^{\tau} \left(x'[t]\,\hat{Q}\,x[t] + u'[t]\,\hat{R}\,u[t]\right) dt\right\} \tag{2.3}$$

where $E$ represents the expected value operation and the weighting matrices $\hat{R}$ and $\hat{Q}$ satisfy $\hat{R} > 0$ and $\hat{Q} \geq 0$. Because of the time-averaging nature of the performance index, this LQG problem is called the *steady-state* LQG problem [1].

The control objective will be to minimize the index $J_c$ with a discrete-time linear compensator as shown in the configuration of figure 2-1, where the input u[t] is now piecewise constant. The solution to this problem involves discretizing the plant model and performance index, and then solving the resulting discrete-time LQG problem. Discretizing the equations (2.1)-(2.3) for a sampling period of $T$

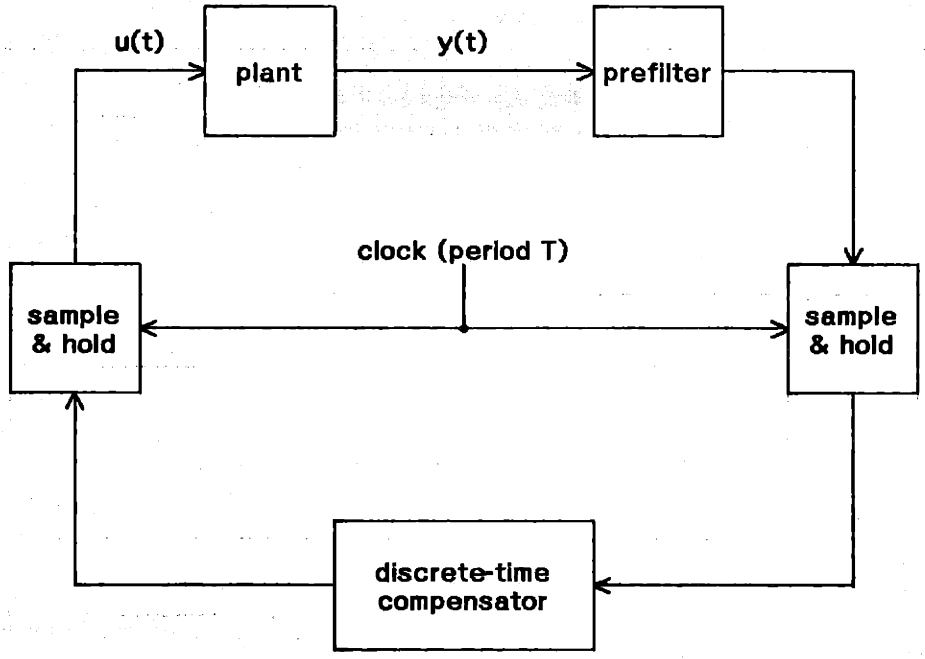



Figure 2-1: LQG Configuration

seconds produces: [1,25,26]

$$x(k+1) = \Phi\, x(k) + \Gamma\, u(k) + w_1(k) \tag{2.4}$$

$$y(k) = L\, x(k) + w_2(k) \tag{2.5}$$

$$J_d = E\left\{ \lim_{l\to\infty} \frac{1}{2l} \sum_{k=-l}^{l} \left( x'(k)\, Q\, x(k) + 2x'(k)\, M\, u(k) + u'(k)\, R\, u(k) \right) \right\} \tag{2.6}$$

Note the inclusion of the cross-term weighting matrix $M$ in (2.6). Equations (2.4) and (2.5) describe the behavior of the plant at the sample times, and the index $J_d$ in (2.6) satisfies:

$$\lim_{T \to 0} J_d = J_c \tag{2.7}$$

Equation (2.7) does not imply that $J_d$ decreases monotonically towards $J_c$ as $T$ approaches zero. In fact, for systems that are open-loop oscillatory, $J_d$ will be near-infinite if $T$ is an integer multiple of the period of the oscillation [25].

The discrete-time parameters in (2.4)-(2.6) are defined as follows:

$$\Phi(\tau) = e^{A\tau}; \qquad \Gamma(t) = \int_0^t \Phi(\tau) B \, d\tau$$

$$\begin{aligned}
&\Phi = \Phi(T) \\
&\Gamma = \Gamma(T) \\
&L = C \\
&Q = \frac{1}{T} \int_0^T \Phi'(\tau) \hat{Q} \, \Phi(\tau) \, d\tau \\
&R = \hat{R} + \frac{1}{T} \int_0^T \Gamma'(\tau) \hat{Q} \, \Gamma(\tau) \, d\tau \\
&M = \frac{1}{T} \int_0^T \Phi'(\tau) \hat{Q} \, \Gamma(\tau) \, d\tau
\end{aligned} \tag{2.8}$$

The discrete uncorrelated white noise vectors $w_1(k)$ and $w_2(k)$ have the following covariance matrices:

$$\begin{aligned}
&\Theta_1 = \int_0^T \Phi(\tau) \Xi_1 \Phi'(\tau) \, d\tau \\
&\Theta_2 = \frac{1}{T} \Xi_2
\end{aligned} \tag{2.9}$$

The factor of $\frac{1}{T}$ in the expression for $\Theta_2$ arises from the filter preceding the output sampler in figure 2-1. Such a lowpass filter (of bandwidth $\frac{2}{T}$) will be assumed

to pass the signal $Lx$ unchanged, while filtering the white measurement noise $w_2$. Due to the fictitious nature of white noise (its unlimited bandwidth), one cannot actually sample it unfiltered without obtaining a sample of infinite variance. (Aliasing [28,29,30], which is an overlapping of the spectrum of the sampled signal, would cause the infinite variance.)

The solution to the discrete-time LQG problem, given in Sage [27], gives rise to the following ideal compensator:

$$\begin{aligned} \hat{x}(k+1) &= \Phi\hat{x}(k) + K\left(y(k+1) - L\Phi\hat{x}(k)\right) + \Gamma u(k) \\ u(k+1) &= -G\hat{x}(k+1) \end{aligned} \tag{2.10}$$

where $\hat{x}$ represents the state estimate, $G$ is computed off-line as the solution to an optimal regulator problem, and $K$ is computed off-line as the solution to a Kalman filter problem.

Immediately, a problem arises in trying to implement the compensator described in (2.10). The system shown in figure 2-1 and equations (2.4)-(2.6) assumes that the output and input samplers operate simultaneously. However, equations (2.10) clearly show a dependence of $u(k+1)$ on $y(k+1)$. Since it takes a finite amount of time $t_c$ to compute $u(k+1)$ *after* $y(k+1)$ is present at the sampler output, $u(k+1)$ cannot be generated until some time *after* the $(k+1)^{th}$ sample time. This contradiction makes it impossible to implement (2.10) as described.

Such a problem is easy to avoid once recognized. One way to get around the contradiction is simply to delay the clock driving the zeroth-order hold at the compensator output by $t_c$ seconds. Leaving all else the same, this approach will give approximately the right result whenever $t_c << T$. However, a more general

procedure that will work for any $T \geq t_c$ is desirable.

Kwakernaak and Sivan [1] have presented such a design method, *including* the possibility of calculation delay in the initial design. This procedure involves two steps. First, to ensure that the compensator can be physically implemented, we must restrict the control $u(k)$ to depend only on observations up to and including $y(k)$ — *not* $y(k+1)$. However, if the calculator time $t_c$ is much less than the sample period $T$, this presents some inefficiency, since the new value $u(k+1)$ is available (the computations are completed) long *before* it is needed as input to the hold unit. Thus Kwakernaak and Sivan also allow for a *delaying* of the clock driving the system output ($y$) sample-and-hold unit by a time $\delta$ relative to the clock driving the system input ($u$) zeroth-order hold (*sample skew*, see figure 2-2). Thus the plant state is discretized at times $kT$ and the output at times $kT+\delta$, although each of these samples will be referred to with '$k$' in the discrete model. The terms '$x(k)$' and '$y(k)$' will no longer represent $x$ and $y$ at the identical instant. This fact must be reflected in the discrete-time model equations [1, Section 6.2].

The expression for $y(kT+\delta)$ can be written using the variation-of-constants formula:

$$y[kT+\delta] = Ce^{A\delta}x[kT] + w_2[kT+\delta] + \int_{kT}^{kT+\delta} e^{A(kT+\delta-\tau)}\left(B\,u[\tau] + w_1[\tau]\right)d\tau$$

$$= Ce^{A\delta}x[kT] + w_2[kT+\delta] + \left\{\int_0^{\delta} e^{A(\delta-\tau)}d\tau\right\}B\,u[kT] + \int_0^{\delta} e^{A(\delta-\tau)}w_1[\tau]\,d\tau \quad (2.11)$$

In its discrete-time form:

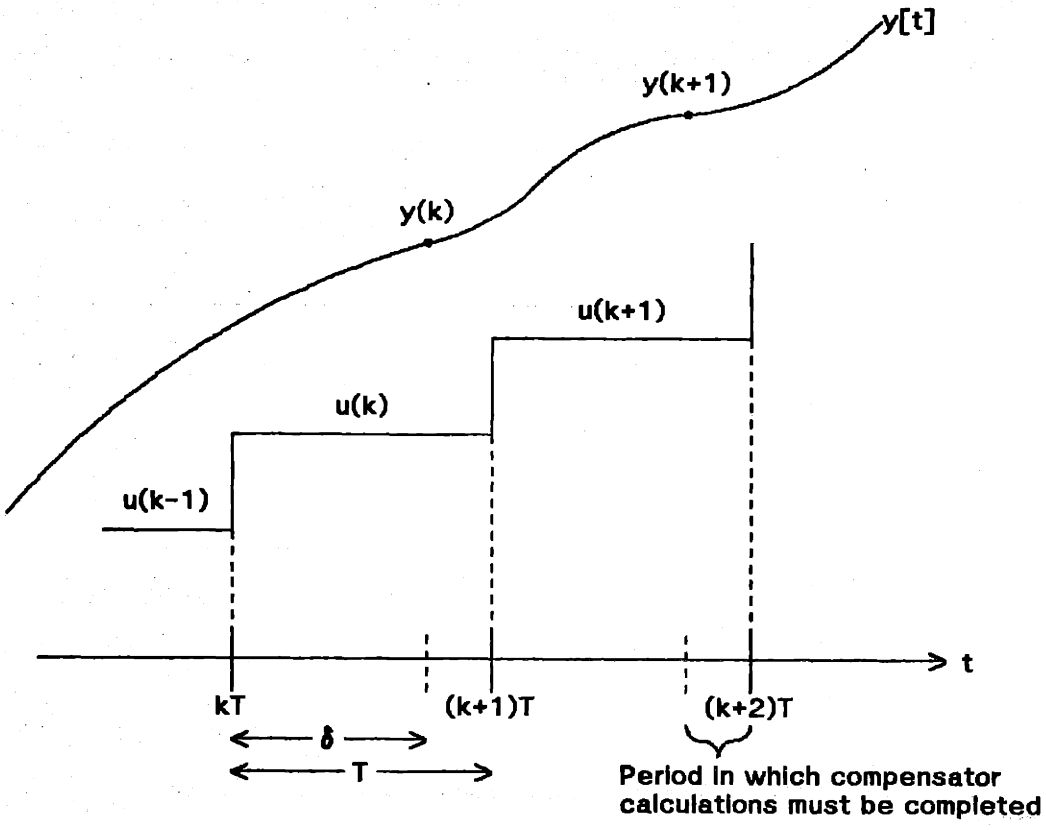



Figure 2-2: From [1], page 523

$$y(k) = L\,x(k) + D\,u(k) + w_2(k) \tag{2.12}$$

where

$$L = C\,\Phi(\delta)$$
$$D = \Gamma(\delta)\,B$$

$$w_2(k) = w_2[kT+\delta] + \int_0^{\delta} \Phi(\delta-\tau)\,w_1[\tau]\,d\tau$$

Model equation (2.12) must replace (2.5). Two complications have been introduced: the feedthrough term $Du(k)$, and the nature of the noise $w_2(k)$. The

noise vectors $w_1(k)$ and $w_2(k)$ have become correlated due to the difference between the input and output clock phases.

$$E\left\{\begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}\begin{bmatrix} w_1'(k) & w_2'(k) \end{bmatrix}\right\} = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{12}' & \Theta_{22} \end{bmatrix} \delta_{kl} \qquad (2.13)$$

where:

$$\delta_{kl} = \begin{cases} 1 & \text{for } k = l \\ 0 & \text{otherwise} \end{cases}$$

$$\Theta_{11} = \int_0^T \Phi(\tau)\,\Xi_1\,\Phi'(\tau)\,d\tau$$

$$\Theta_{22} = \frac{1}{T}\Xi_2 + \int_0^\delta \Phi(\tau)\,\Xi_1\,\Phi'(\tau)\,d\tau$$

$$\Theta_{12} = \int_0^\delta \Phi(T-\tau)\,\Xi_1\,\Phi'(\delta-\tau)\,d\tau$$

By restricting $\hat{x}(k+1)$, or equivalently $u(k+1)$, to depend on the observations up to and including $y(k)$ only, the optimal compensator equations are also modified: [1]

$$\begin{aligned} \hat{x}(k+1) &= \Phi\,\hat{x}(k) + \Gamma\,u(k) + K\left(y(k) - L\,\hat{x}(k) - D\,u(k)\right) \\ u(k+1) &= -G\,\hat{x}(k+1) \end{aligned} \qquad (2.14)$$

where $K$ is the (steady-state) new optimal filter gain matrix ($n \times p$) and $G$ is the optimal regulator gain matrix ($m \times n$). These matrices satisfy discrete algebraic Riccatti equations that can be derived from [1] for the discretized plant and compensator described in (2.4),(2.6),(2.8), and (2.12)-(2.14): (equation (2.15) is also presented in [25].)

$$\bar{P} = \left(\Phi - \Gamma R^{-1} M'\right) \bar{P} \left(\Phi - \Gamma G\right) + Q - M R^{-1} M' \qquad (2.15)$$

where $G = \left(R + \Gamma' \bar{P} \Gamma\right)^{-1} \Gamma' \bar{P} \left(\Phi - \Gamma R^{-1} M'\right) + R^{-1} M'$

and

$$\Sigma = \left(\Phi - K L\right) \Sigma \Phi' + \Theta_{11} - K \Theta_{12}' \qquad (2.16)$$

where $K = \left(\Phi \Sigma L' + \Theta_{12}\right) \left(\Theta_{22} + L \Sigma L'\right)^{-1}$

With this formulation, the compensator (2.14) can be actually implemented so long as $0 \le \delta \le T - t_c$, since the time between the reception of $y(k)$ and the generation (sampling) of $u(k+1)$ must be long enough (at least $t_c$ seconds) to complete the computations involved. Whenever the calculation time is comparable to the sample period, or the sample rate is much greater than the system bandwidth, it is advantageous to choose $\delta$=0. Such a choice simplifies (2.16) since $\Theta_{12}$=0, allows for a simpler hardware clocking arrangement for the samplers, and can also reduce the on-line computation time $t_c$ since $D$=0. For the examples treated in this thesis, $\delta$ will be assumed to be zero for simplicity. The results easily extend to the non-zero $\delta$ case.

In this study, only single-input single-output plants will be considered ($m$=$p$=1). With this choice, we can naturally build on the existing digital filtering results, and still bring out the issues we wish to discuss. Consideration of the multiple-input multiple-output case would raise even more issues, and probably obscure the points we wish to make. Even in digital signal processing, there are very few multiple-input multiple-output results. The extension of our results for

control systems to the multiple-input multiple-output case would be valuable, and in most cases, is not too difficult. Topics such as multiple-input scaling, multiple-output pipelining, and multi-loop limit cycles are discussed in some detail in the closing chapter of this thesis.

# Chapter 3: Compensator Structures

## §3.1 Introduction

Chapter 2 has described the background and basic derivation of the LQG compensator. The net result was the set of equations (2.14). Since the plant is connected to the compensator at only two points, $u$ and $y$, the ideal compensator can be completely described by an input-output map, or transfer function (recall that we are concentrating on the single-input single-output case). In terms of the parameters in (2.14), this transfer function is written:

$$H(z) = \frac{U(z)}{Y(z)} = -G\,(z - \Phi + K\,L + \Gamma\,G)^{-1}K \tag{3.1}$$

When expressed as a ratio of polynomials, (3.1) would have the form (3.2) where $n$ is the order of the plant and thus of the LQG compensator. The lack of a term $a_0$ in the numerator follows from the dependence of $u(k)$ only on past values of $y$, as explained in Chapter 2.

$$H(z) = \frac{a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}} \tag{3.2}$$

Equation (3.1) or (3.2) represents the *ideal* discrete-time response of the LQG compensator. Note that in these transfer functions, $y$ represents the compensator input and $u$ the output, which is the reverse of the filtering case typically considered in digital signal processing.

Now consider that (3.1) or (3.2) is to be implemented *digitally* (as a digital network, or filter [31]). Figure 3-1 presents a simple block diagram of this system. The transfer function (3.1) must now be implemented infinite precision with

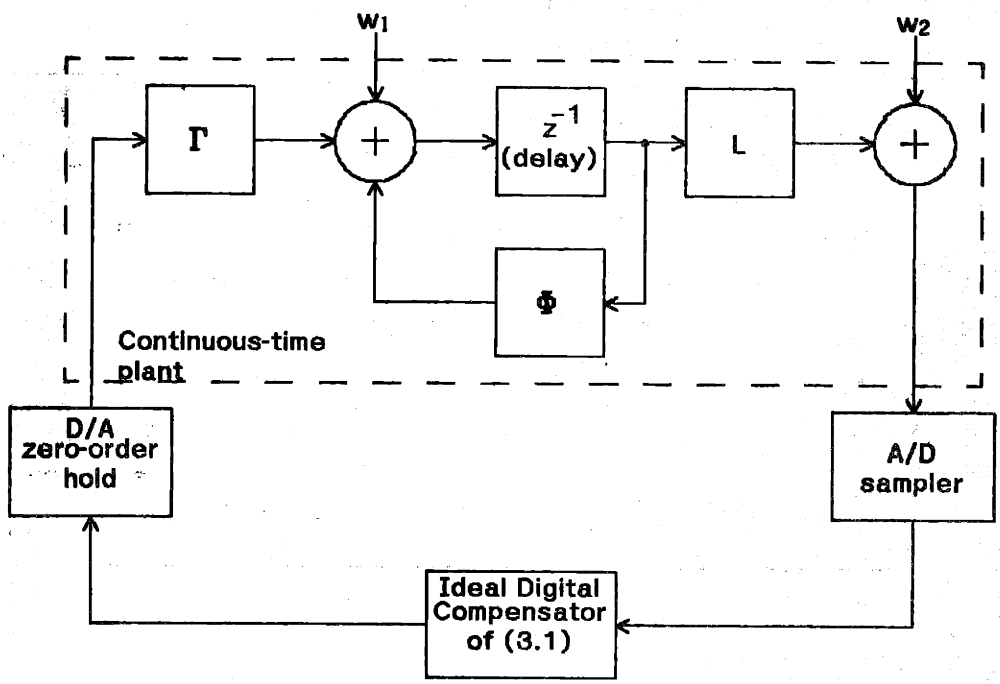



Figure 3-1: Plant And Digital Compensator

as little degradation in some system performance measure as possible, subject to certain constraints on the speed and cost of the attendant hardware. In the setting of a steady-state LQG problem, it is convenient to select the performance index J in (2.6) as the measure of performance, since it reflects the weighted steady-state root-mean-square state and control fluctuations. It would also have been possible to choose a criterion such as phase margin, output noise power, or any combination of stability or noise measures. If the problem under consideration were simply a Kalman filter, then a suitable performance measure would be the trace of the error covariance matrix. We have chosen *J* in order to present our results in a specific context. These results extend in a simple and direct fashion to the error covariance trace, and with more difficulty to phase margin and gain

margin measures.

In this chapter, we will discuss the concept of *structures* for digital compensators, and examine accurate ways of representing the arithmetic operations implicit in such structures. Adapting the results of digital signal processing, we will develop an accurate notation for compensator structures. Several classes of structures will then be presented using this new notation.

## §3.2 Structures and Notation

As explained in Chapter 1, the term *Implementation* includes the choice of a suitable structure to approximate (3.1) (or (3.2)) assuming fixed-point arithmetic, and the specification of the hardware architecture and components. This section will adapt digital filtering concepts to develop structures for digital compensators and to formulate an accurate notation for these structures. The state space form common in control applications will be shown to be inadequate for this purpose.

The term *structure* will be employed to specify the exact finite-precision mathematical procedure by which the compensator output samples $u$ are generated from its input samples $y$. All structures for implementing a given filter or compensator would perform identically under infinite-precision arithmetic, but will produce different quantization noise, coefficient quantization effects, and limit cycles given the (realistic) finite-precision environment.

Consider a very simple example. Assume that an ideal compensator has been designed, and that its (infinite-precision) transfer function is:

$$H(z) = \frac{z^{-1}}{1 + 1.11z^{-1} + 0.287z^{-2}} \qquad (3.3)$$

Figure 3-2a shows a *signal flow graph* [28,29] of one possible structure, the

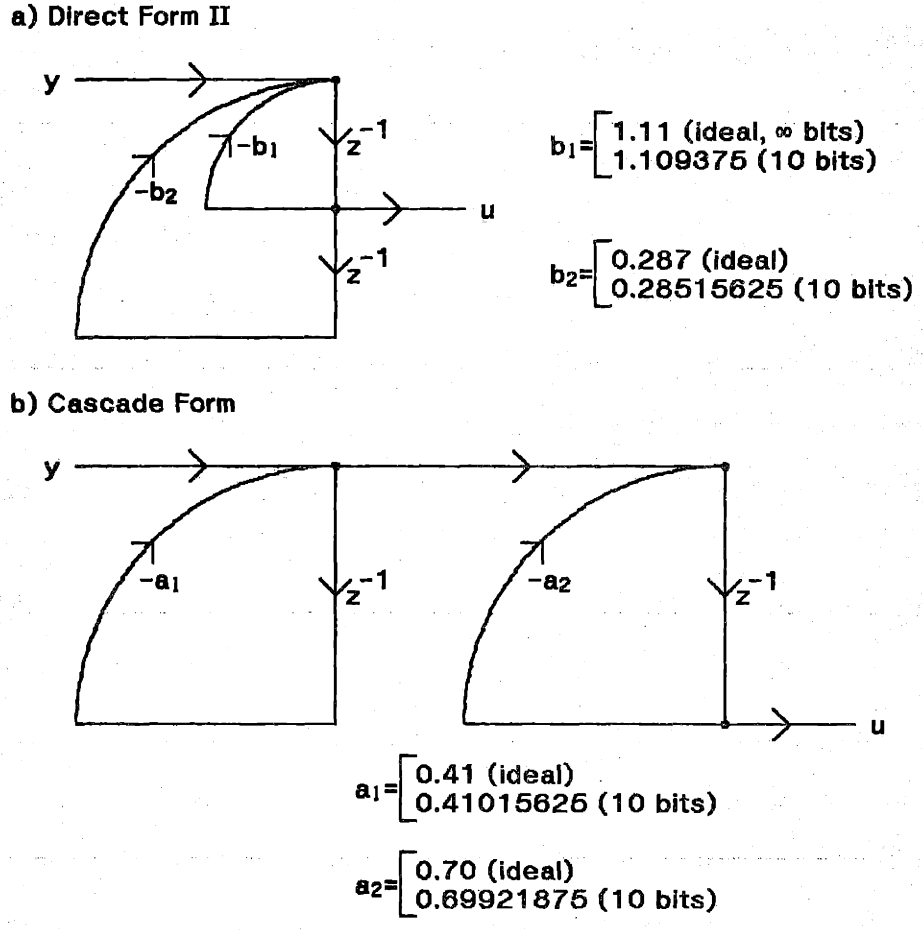



Figure 3-2: Example Structures

direct form II [28], for implementing (3.3). The infinite-precision values for $b_1$ and $b_2$ can be read directly from (3.3). Given only 10-bit coefficient registers, these values must be quantized (assume rounding). Reserving one bit for the integral portion of the coefficient word (bits to the left of the binary point), one

sign bit, and 8 bits for the fractional portion, the rounded coefficient values would be 1.109375 and 0.28515625.

Figure 3-2b shows the flow graph of another common structure, the cascade form. Here we realize (3.2) with a cascade of two first-order filter sections. The coefficients $a_1$ and $a_2$ can be found by factoring the denominator of (3.2). Again, the ideal values must be rounded to fit 10-bit words, producing $a_1$ = 0.69921875 and $a_2$= 0.4105625.

Now let us examine the performance of these two structures given their respective finite-precision coefficients. The (10-bit) direct form II and the cascade have the transfer functions shown in (3.4) and (3.5) respectively:

$$H(z) = \frac{z^{-1}}{1 + 1.109375z^{-1} + 0.28515625z^{-2}} \tag{3.4}$$

$$H(z) = \frac{z^{-1}}{1 + 1.109375z^{-1} + 0.2867889404296875z^{-2}} \tag{3.5}$$

Clearly these two structures produce slightly different transfer functions under finite precision, and we have not even considered their respective quantization noise and limit cycle behavior. Thus different structures will in general result in different finite-precision performance, even though their infinite-precision counterparts have equivalent performance (that of the ideal design).

In order to discuss or analyze *different* implementation structures, one must have a notation (other than the pictorial signal flow graph) that accurately reflects these differences. From the system theoretic approach, it seems natural to examine the discrete-time state space representation for a digital filter (with input $u$ and output $y$):

$$\begin{aligned} v_{k+1} &= \Psi_{11} v_k + \Psi_{12} u_k \\ y_k &= \Psi_{21} v_k + \Psi_{22} u_k \end{aligned} \tag{3.6}$$

In this representation, the states $v$ are defined to be the outputs of the delay elements in a signal flow graph, and the multiplier coefficients in $\Psi_{11}$, $\Psi_{12}$, $\Psi_{21}$, and $\Psi_{22}$ are the gains between state or input nodes and next-state or output nodes.

Unfortunately, while this form of notation does accurately represent a class of structures, it is not sufficiently general to represent the arithmetic operations associated with any structure. This lack of generality arises in representing structures whose signal flow graphs must have *intermediate* nodes, that is, nodes which are not state nodes or the input or output node. Figure 3-3 presents such a

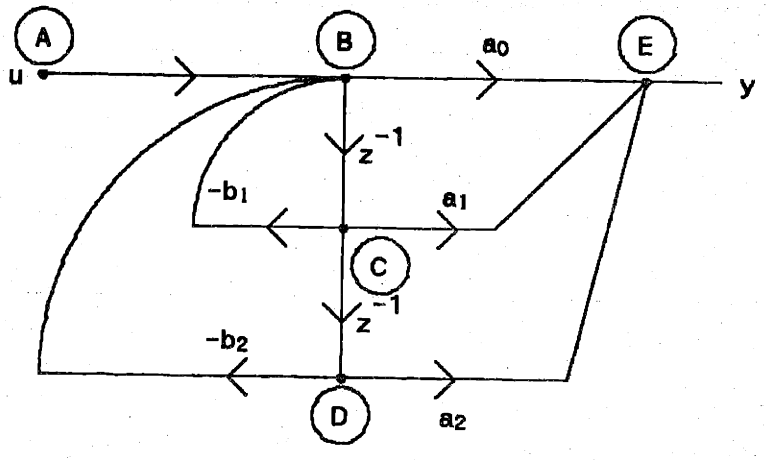



Figure 3-3: Example Structure

structure, a two-pole two-zero direct form II structure. Nodes #C and #D are state nodes, node #A is the input node, and node #E is the output node. However, the $a_0$ branch begins at an intermediate node, node #B. Thus there would be

no way to include the coefficient $a_0$ as an entry in any of the state space matrices $\Psi_{11}$, $\Psi_{12}$, $\Psi_{21}$, or $\Psi_{22}$. From another viewpoint, the state space representation lacks any way of expressing the implicit ordering, or *precedence*, associated with the operations involved in certain filter structures. For state space representations, all multiplications can occur at once (independently), and then all additions can occur. For the direct form II structure of figure 3-3, the multiplications $-b_1$, and $-b_2$ must precede the addition at node #B which must then precede the following multiplication by $a_0$. This sequence of operations cannot be adequately expressed by equations of the form (3.6). This point is clearly illustrated by Willsky [16], pages 122-124.

At this point it is convenient to turn to the field of digital signal processing for an adequate way to represent filter structures. Crochiere [31,32] has described matrix equations for correctly computing the node signal values in any filter structure. Let the signal value at the $i^{th}$ node (of $N_0$ nodes) at time $k$ be $y_i(k)$ and the external input to node $i$ be $u_i(k)$. Between any two nodes $i$ and $j$ there can exist one interconnecting branch of constant gain $F_{cij}$, and/or one multiply-and-delay branch $F_{dij}$. These branches and their interconnected nodes form an *elementary* network. (We have further assumed that all values $F_{dij}$ are either zero or one, with no loss of generality.) For an elementary network then, the node value $y_i(k)$ may in general depend on all node values at time $k-1$ and some of the node values at time $k$, depending on whatever branches exist:

$$y_i(k) = \sum_{j=1}^{N_0} F_{cji}\, y_i(k) + \sum_{j=1}^{N_0} F_{dji}\, y_i(k-1) + u_i(k) \qquad (3.7)$$

Thus $F_c$ is an $N_0 \times N_0$ matrix of constant-branch coefficients, and $F_d$ is an $N_0 \times N_0$ matrix of delay-branch coefficients. In most networks, a substantial number of the entries in $F_c$ and $F_d$ are zero, and as stated above, the remaining entries in $F_d$ are ones. In $z$-transform notation, the vector quantity $Y(z)$ can be written:

$$Y(z) = U(z) + F_c' Y(z) + F_d' Y(z)\, z^{-1} \qquad (3.8)$$

The transfer function matrix $H(z)$ defined by $Y(z) = H'(z)\, U(z)$ can be derived from (3.8):

$$H(z) = \left[ I - F_c - F_d\, z^{-1} \right]^{-1} \qquad (3.9)$$

Now lets take a look at computing the node signal values. These calculations must occur between the time instants $k-1$ and $k$. Some of the node updates will involve the past values at time $k-1$, and some will involve already-updated values. Thus the node values must be computed in the proper order. For example, the first node value to be updated should not depend on any other updated node values, since these would not yet have been computed. Thus in terms of the matrix notation above, a correct *node precedence*, or ordering, would only depend on the constant-coefficient branches, since all delayed values $y(k-1)$ are known at time $k$. Crochiere [32] describes a formal node-ordering technique:

(1) All nodes entered by inputs or delay branches *only* are placed in node class 1.

(2) Remove from the network all class 1 nodes and any branches connected to them.

(3) Repeat steps 1 and 2 on the remaining network, for node classes 2, 3, ··· until all nodes are classified.

(4) Order from 1 to $N_0$ all nodes, first using all the class 1 nodes, then class 2 and so on.

This technique will not result in a unique ordering of the nodes, but the ordering produced will satisfy the above-mentioned computational constraints.

If this ordering procedure can be carried out, the digital network, or structure, is *computable*, and the resulting $F_c'$ matrix is zero on and above the main diagonal. If not, the network had at least one closed loop without delay, and does not represent an implementable structure. Note that a non-recursive structure has an ordering whereby $F_c'$ is also zero on and above the main diagonal.

As an example of this matrix signal-flow-graph formulation, consider the five-node structure of figure 3-3. Using the ordering algorithm presented above, nodes #C and #D fall into class 1, node #A falls into class 2, node #B into class 3, and node #E into class 4. Thus we can define nodes #1 through #5 with the ordering *C,D,A,B,E*. The following five equations now define the (frequency) response of the network:

$$
\begin{aligned}
Y_1 &= & & & z^{-1}Y_4 & & \\
Y_2 &= z^{-1}Y_1 & & & & & \\
Y_3 &= -b_1Y_1 & -b_2Y_2 & & & +U_3 & \\
Y_4 &= & & Y_3 & & & \\
Y_5 &= a_1Y_1 & +a_2Y_2 & & +a_0Y_4 & &
\end{aligned}
\tag{3.10}
$$

The 5×5 matrices $F_c$ and $F_d$ can be formed using (3.8) and (3.10), and the resulting matrix $H(z)$ is given in (3.11):

$$
H(z) = \begin{bmatrix} 1 & -z^{-1} & b_1 & 0 & -a_1 \\ 0 & 1 & b_2 & 0 & -a_2 \\ 0 & 0 & 1 & -1 & 0 \\ -z^{-1} & 0 & 0 & 1 & -a_0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \tag{3.11}
$$

For a single-input single-output digital filter such as the one in figure 3-3, we specify only the scalar input-output map $H_{ij}(z)$. ($H_{35}(z)$ in the example above) The remaining entries of $H(z)$ represent transfer functions from or to nodes that are internal to the structure.

A deficiency of the matrix notation above appears when we consider structural *transformations*. Such transformations are very useful in generating new structures with identical infinite-precision transfer functions as some original structure, but with different finite-precision performance. For a structure which can be accurately represented with state space notation, the similarity transform fills this role. For the Crochiere matrix representation, a transformation technique also exists [17]. This technique must be constrained so that the transformed structure is computable; in other words, it must have no delay-free loops [17]. However, even with this restriction, the number of delay branches and the degree of pre-

cedence inherent in the additions and multiplications of the resulting (infinite-precision-equivalent) structure are in general *unpredictable.*

To combat this difficulty, a notation as convenient and useful for transformation as the state space form, but with the generality of the Crochiere matrix representation is desirable. Such a notation, related to the state space notation, has been presented by Chan [17]. As in a state space, define the outputs of delay elements to be the states $v$, and let $y$ be the filter or compensator input and $u$ be the output. Then the coefficients and the sequence of multiplications and additions in *any* filter structure can be specified with the following representation:

$$\begin{bmatrix} v(k+1) \\ u(k) \end{bmatrix} = \Psi_q \, \Psi_{q-1} \cdots \Psi_1 \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \tag{3.12}$$

where $\Psi_q, \cdots, \Psi_1$ are matrices representing the arithmetic and quantization operations in the structure. Three important points make (3.12) useful:

(1) Each (rounded) coefficient in the structure occurs once and only once as an entry in one of the $\Psi_l$ matrices. The remainder of the matrix entries are ones and zeros.

(2) All intermediate *(non-storage)* nodes in a structure are represented in the vectors $r_1(k) = \Psi_1 \begin{bmatrix} v(k) \\ y(k) \end{bmatrix}$, $r_2(k) = \Psi_2 r_1(k)$, ..., $r_{q-1}(k) = \Psi_{q-1} r_{q-2}(k)$. This point is especially important since both the state nodes $v$ and intermediate nodes $r$ must be scaled to satisfy dynamic range constraints. (See Chapter 5).

(3) The concept of *precedence* for the operations (multiplies, adds, and quantizations) is maintained. The ordering of the $\Psi_l$ matrices implies that the

operations involved in computing $r_1(k)$ are completed first, then $r_2(k)$ next, and so forth. Thus the matrix $\Psi_q$ contains the operations of lowest precedence, and the parameter $q$ specifies the number of *precedence levels.*

Consider the example of figure 3-2. Using the procedure outlined in Chan [17], the direct form II structure in figure 3-2 has a one-level representation as shown in (3.13), while the cascade structure of figure 3-2 requires two levels to describe its operations (3.14).

$$\begin{bmatrix} v(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -b_2 & -b_1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \tag{3.13}$$

$$\begin{bmatrix} v(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -a_2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -a_1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \tag{3.14}$$

It should be noted here that the representations shown in (3.13) and (3.14) are not unique. The numbering of the intermediate nodes $r$ (within precedence constraints) is arbitrary. (This nonuniqueness is also true of the Crochiere matrix representation, since node numbering within a class is arbitrary.) Furthermore, some of the $r$ nodes are trivial, as can be seen by reversing the procedure and generating a structure directly from (3.14) — see figure 3-4. Nodes $r_{11}(k)$ and $v_1(k+1)$ are equivalent nodes, separated only by a trivial multiplication by one in $\Psi_2$. The same is true of $r_{12}(k)$ and $v_2(k)$. Figure 3-2b is simply a *node-minimal* version of figure 3-4 [17,32]. However, *all such representations are equivalent in terms of their finite-precision behavior* — they all effectively represent the same

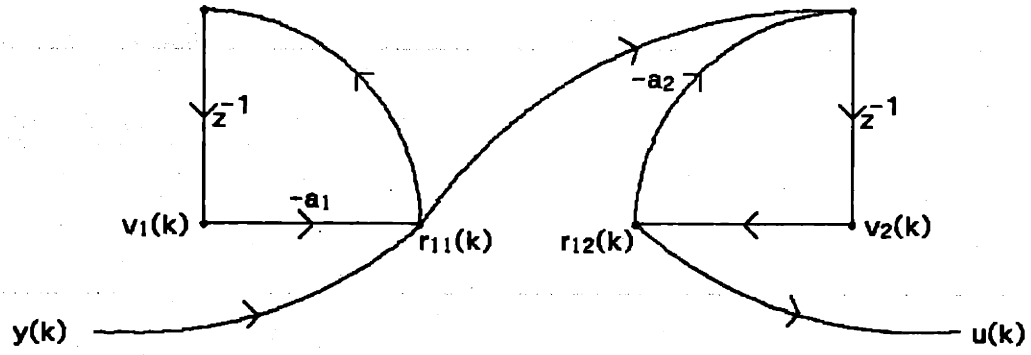



Figure 3-4: Exact Structure of (3.14)

structure [17,32]. It does not matter which is chosen.

In terms of its generality, the notation described by Chan is as useful as the Crochiere representation. In fact, Chan presents a technique for converting from any (elementary) signal flow graph to his state space-related notation, and then back again to an equivalent signal flow graph. Also, in the context of Chan's notation, we can now see that a state space will represent only a *class* of structures, namely those with only one inherent level of precedence.

An important advantage to the notation introduced by Chan is the ease with which transformations [17] can be applied to generate new structures that are infinite-precision-equivalent to some original structure. This technique is an adaption of the similarity transformation used with a (one-level) state space. Define:

$$\tilde{\Psi}_l = P_l \, \Psi_l \, P_{l-1}^{-1} \qquad \text{for } l = 1, \cdots, q \tag{3.15}$$

where the $P_l$ for $l = 1, \cdots, q-1$ are general non-singular transformation matrices of appropriate dimension and

$$P_0 = \begin{bmatrix} P & 0 \\ 0 & 1 \end{bmatrix} : \quad P_q = \begin{bmatrix} P & 0 \\ 0 & 1 \end{bmatrix} \tag{3.16}$$

The new (transformed) structure will then have the following representation:

$$\begin{bmatrix} \tilde{v}(k+1) \\ u(k) \end{bmatrix} = \tilde{\Psi}_q \tilde{\Psi}_{q-1} \cdots \tilde{\Psi}_1 \begin{bmatrix} \tilde{v}(k) \\ y(k) \end{bmatrix} \tag{3.17}$$

What makes this transformation method so useful is that the original and transformed structures have the *same* number of states (delays) and the *same* number of precedence levels. It is also possible to restrict the matrices $\{P_0, P_1, \cdots, P_q\}$ to control the number of non-unity, non-zero entries in the new $\tilde{\Psi}$ matrices, as explained in Chapter 8.

Now let us try to apply this valuable notation to represent structures for digital feedback compensators. Unfortunately, the notation described by Chan is not quite adequate for the control setting. To demonstrate this point let us consider the direct form II structure in figure 3-3 as a compensator structure. In the notation of Chan, this structure will have the following representation:

$$\begin{bmatrix} v(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -b_2 & -b_1 & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} \tag{3.18}$$

According to this set of equations, the *next-state* vector $v(k+1)$ is a function of the *present* state and input (no problem). However, (3.18) also describes the current *output* to be a function of the current state and input. From the viewpoint of causality this expression must be in error, since some finite amount of time is needed for the computation of $u(k)$ *after* $v(k)$ and $y(k)$ are generated. In most digital filtering applications, a short delay in obtaining the output (series delay) is

of no concern, and hence the representation in (3.18) *is* adequate for filters. However, in control applications, such delay is critical since the filter is embedded in a feedback loop (recall Chapter 2). Our approach must reflect the true operation of the compensator, accounting for all necessary computational delays.

One simple approach to solving this problem might be to include the delay as an explicit series delay *following* the compensator described in Chan's notation. Unfortunately, this implies that the delay be included as part of the control system plant. Thus every LQG design would involve initially augmenting the plant, and then designing an optimal LQG compensator. For an $n^{th}$-order system, this procedure creates an $(n+1)^{th}$-order augmented system, and thus an $(n+1)^{th}$-order compensator. Clearly this approach has a disadvantage; it increases the compensator order. Furthermore, by not including the extra delay in some way *within* the compensator itself, we may have restricted the types of structures that are possible for compensator implementations.

Thus we must search for a better approach. Let us include the extra delay *within* the compensator structure itself. The compensator design technique of Chapter 2 ensures that the output $u(k)$ depends on *past* inputs, not present inputs. (This seems to force the compensator to include an entire *sample* delay time $T$ instead of simply a calculation time delay, which may be considerably shorter. However, recall the sample-skew issue discussed in Chapter 2.) Thus we can represent $u(k+1)$ as a function of $v(k)$ and $y(k)$, rather than as a function of $v(k+1)$ and $y(k+1)$. Then $u(k)$ can be generated by a unit delay following $u(k+1)$. The node $u(k)$ thus becomes an additional compensator *state*. In terms of adapting the notation of Chan, let us choose $u(k)$ to be the last state (numerically), and write:

$$\begin{bmatrix} v(k+1) \\ u(k+1) \end{bmatrix} = \Psi_q \, \Psi_{q-1} \cdots \Psi_1 \begin{bmatrix} v(k) \\ u(k) \\ y(k) \end{bmatrix} \tag{3.19}$$

where the vector $v$ and also the scalar $u$ are the states of the structure (outputs of delay elements). Thus we have slightly altered the notion of a structure for compensators. Unlike filter structures, $u(k)$ is always both an output *and* a state. The notation in (3.19) for describing compensator structures will be called its *modified state space representation*.

The major implication of this adaptation is that $n^{th}$-order compensators will now require structures having $n+1$ unit delay elements, rather than $n$ as with digital filters. In addition, certain common digital filter structures (for example, the direct form II and cascade and parallel structures based on it) will no longer appear quite the same when used for digital compensators. Each will have an extra delay at the output node, as compared to the corresponding filter structure. In terms of their modified state space representations, the $\Psi_1$ matrix for such structures will have an all-zero next-to-last column. This must occur whenever the node $u(k)$ does not feedback to the rest of the structure. Section 3.3 will show examples of such structures, and we will still refer to them by their corresponding digital filtering designations — see figures 3-5, 3-6, and 3-8). For the remainder of this thesis, the modified state space of (3.19) will be employed to describe compensator structures, and all signal flow graphs will reflect the delay (state) necessary for $u(k)$.

One final implication of the adapted concept of a structure should be brought out. In terms of the transformation procedure described in (3.15) and (3.16), a change is necessary to accommodate compensator structures. In

(3.16), due to the inclusion of the output as a state, the transformation matrix $P_0$ must now be written $\begin{bmatrix} P & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. The extra row and column in this matrix reflect the modified state space representation, and the unity diagonal entry is necessary since the transformation procedure cannot be permitted to alter the output node.

It is also notationally convenient to define the matrix $\Psi_\infty$. Let the coefficients in each $\Psi_l$ matrix be replaced by their infinite-precision counterparts (their values *before* rounding). Then $\Psi_\infty$ is defined to be the infinite-precision product $\Psi_q \Psi_{q-1} \cdots \Psi_1$. This matrix will be used in the derivations of Chapters 5, 6, and 8.

## §3.3 Classes of Structures

Before discussing some of the various classes of structures that exist, it is important to understand the different points of comparison that should be considered. Beyond the finite wordlength effects of quantization noise, coefficient rounding, and limit cycles that are treated in Chapters 5, 6, and 7, one must compare the number of delay elements, coefficients (multiplications), additions, and precedence levels, and also the number of scalers needed to satisfy dynamic range constraints. We will examine structures that are typically *canonic* (minimal) with respect to the number of delay elements, implying a minimal number of storage registers. In order to present specific examples of structures, let us assume that the plant is sixth order ($n$=6).

Given the transfer function (3.2), the most straightforward structure to

consider is the direct form II [28]. As an LQG compensator structure, its signal flow graph is shown in figure 3-5. It is canonic in delays with 7 (in general, $n$+1), has 12 coefficients (non-unity multipliers) and requires only one additional *scaler*. (Scaling, fully discussed in Chapter 5, involves a *normalization* of the structure so that roundoff noise effects and overflows can be held to a minimum. In this process, some of a structure's coefficients will be altered, including certain *unity* entries. Such unity entries will be called scaling multipliers, or scalers, and indicated in signal flow graphs and equations with an asterisk.) The modified state space representation of the direct form II is given below with its two precedence levels. Note that figure 3-5 includes a rough indication of which operations belong in which precedence level.

$$\Psi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ a_6 & a_5 & a_4 & a_3 & a_2 & a_1 \end{bmatrix} \tag{3.20}$$

$$\Psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -b_6 & -b_5 & -b_4 & -b_3 & -b_2 & -b_1 & 0 & 1^* \end{bmatrix}$$

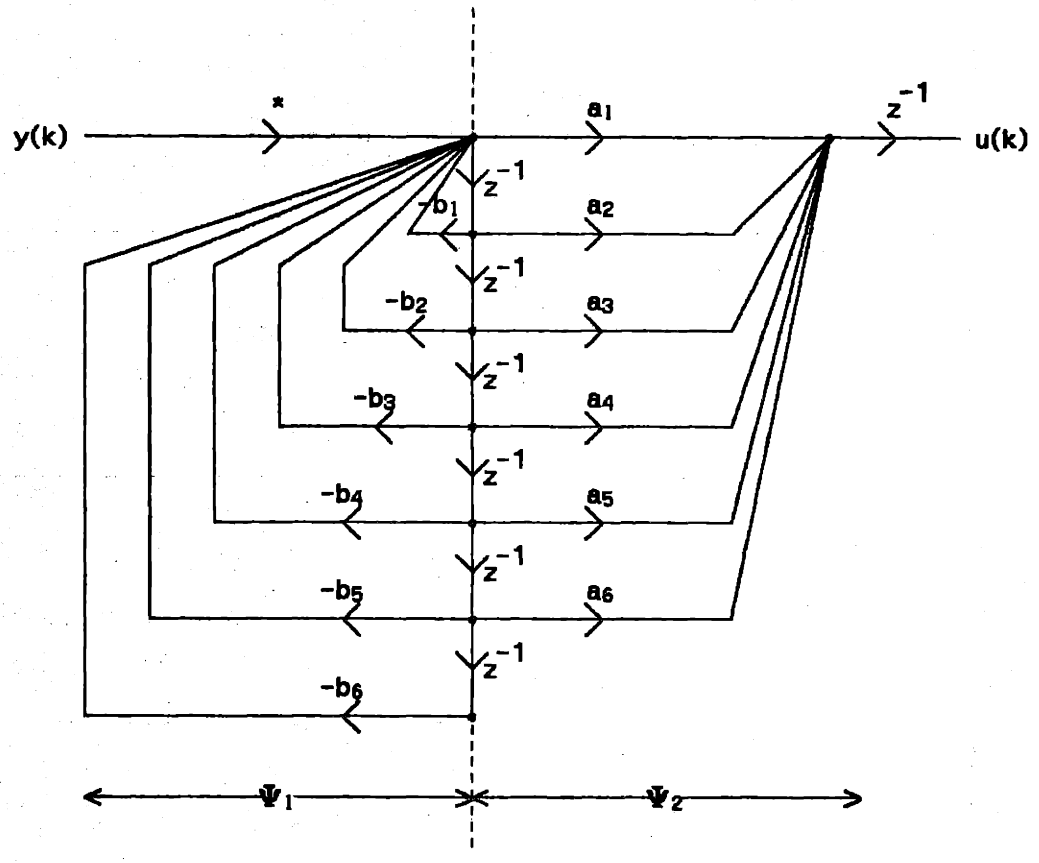



Figure 3-5: Direct Form II Structure (sixth order)

The coefficients in this structure (before scaling) are read directly from the transfer function (3.2).

For higher-order filters, the direct form structure is known to perform poorly in terms of the degradation resulting from the use of finite wordlengths [33]. The dynamic range of the coefficients alone grows with filter order, when the poles are clustered in the *z*-plane. (As shown in Chapters 5 and 6, this will be true for the direct form II compensator structure also.) Consequently, *factored* structures,

such as the *cascade* (of first- and second-order filter sections) are commonly used. This structure is obtained from a multiplicative factoring of the transfer function (3.2):

$$H(z) = \frac{(d_1 z^{-1}+d_2 z^{-2})(1+d_3 z^{-1}+d_4 z^{-2})(1+d_5 z^{-1}+d_6 z^{-2})}{(1+c_1 z^{-1}+c_2 z^{-2})(1+c_3 z^{-1}+c_4 z^{-2})(1+c_5 z^{-1}+c_6 z^{-2})} \qquad (3.21)$$

If each second-order section is implemented as a direct form II structure, then the cascade compensator structure (figure 3-6) also has 12 coefficients and 7

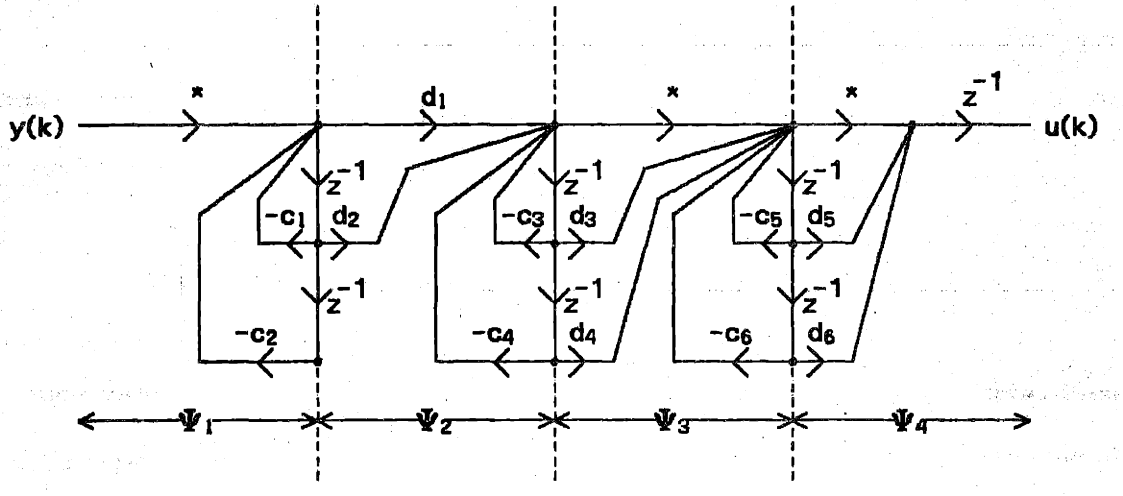



Figure 3-6: Cascade Structure (Direct Form II)

delays (canonic), but requires four precedence levels ($n_s$+1 in general, where $n_s$ is the number of sections) and three scalers:

$$\Psi_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & d_6 & d_5 & 1^* \end{bmatrix}$$

$$\Psi_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & d_4 & d_3 & -c_6 & -c_5 & 1^* \end{bmatrix}$$

$$\Psi_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ d_2 & -c_4 & -c_3 & 0 & 0 & d_1 \end{bmatrix}$$

$$\Psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -c_2 & -c_1 & 0 & 0 & 0 & 0 & 0 & 1^* \end{bmatrix}$$

(3.22)

Actually, this cascade can be used to represent several different structures since the poles and zeros in (3.21) must first be grouped together to form second-order sections, and then the sections must be ordered. Furthermore, the individual sections could be structured in any number of ways (other than the direct form II) [34], [35], each giving rise to a different overall structure.

This variety of second-order sections raises an interesting point. If a cascade or parallel combination of a certain type of section is not delay-canonic when applied to digital filters, it may still be delay-canonic when adapted as a compensator structure. Consider the case of a cascade of direct form I [28] second-order sections. Such a *filter* structure is not delay-canonic (it requires more than $n$ delays). However, due to the added delay used in *compensator* structures, the direct form I compensator structure *is* delay-canonic, requiring $n+1$ unit delay elements. For a sixth-order LQG compensator, such a structure has 7 delay elements and only three (in general $n_s$) precedence levels and two scalers: (See figure 3-7)

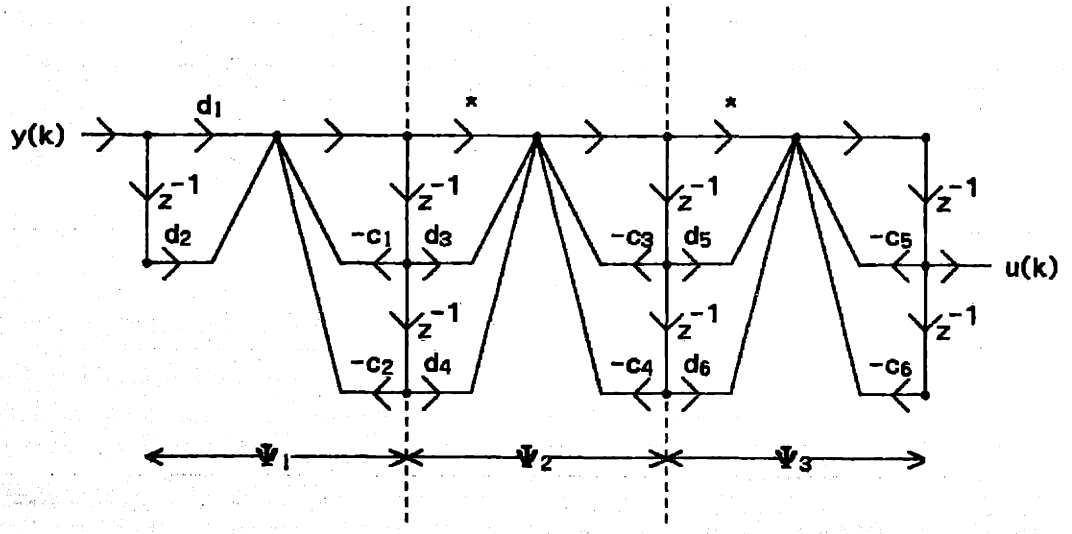



**Figure 3-7: Cascade Structure (Direct Form I)**

$$\Psi_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & d_6 & d_5 & -c_6 & -c_5 & 0 & 0 & 1^* \end{bmatrix}$$

$$\Psi_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ d_4 & d_3 & -c_4 & -c_3 & 0 & 0 & 0 & 1^* \end{bmatrix} \qquad (3.23)$$

$$\Psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ d_2 & -c_2 & -c_1 & 0 & 0 & 0 & 0 & d_1 \end{bmatrix}$$

Another factored form is the *parallel* structure. This structure is obtained from a partial-fraction expansion of (3.2):

$$H(z) = \frac{e_1 z^{-1} + e_2 z^{-2}}{1 + c_1 z^{-1} + c_2 z^{-2}} + \frac{e_3 z^{-1} + e_4 z^{-2}}{1 + c_3 z^{-1} + c_4 z^{-2}} + \frac{e_5 z^{-1} + e_6 z^{-2}}{1 + c_5 z^{-1} + c_6 z^{-2}} \qquad (3.24)$$

Again, using the direct form II for each individual section results in the compensator structure of figure 3-8, which has two precedence levels, 12 coefficients, 7 delays (canonic), and three scaling multipliers:

$$\Psi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ e_2 & e_1 & e_4 & e_3 & e_6 & e_5 \end{bmatrix} \qquad (3.25)$$

$$\Psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -c_2 & -c_1 & 0 & 0 & 0 & 0 & 0 & 1^* \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -c_4 & -c_3 & 0 & 0 & 0 & 1^* \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -c_6 & -c_5 & 0 & 1^* \end{bmatrix}$$

The representation in (3.25) can be used to represent several structures since the real poles, if any, must still be grouped into sections. (The section-ordering and zero-pairing issues of the cascade disappear since all sections are in parallel, and the partial-fraction expansion gives no control over the zero locations.) Also, different types of second-order section structures are possible.

A structure that appears on the surface to be more natural for the LQG

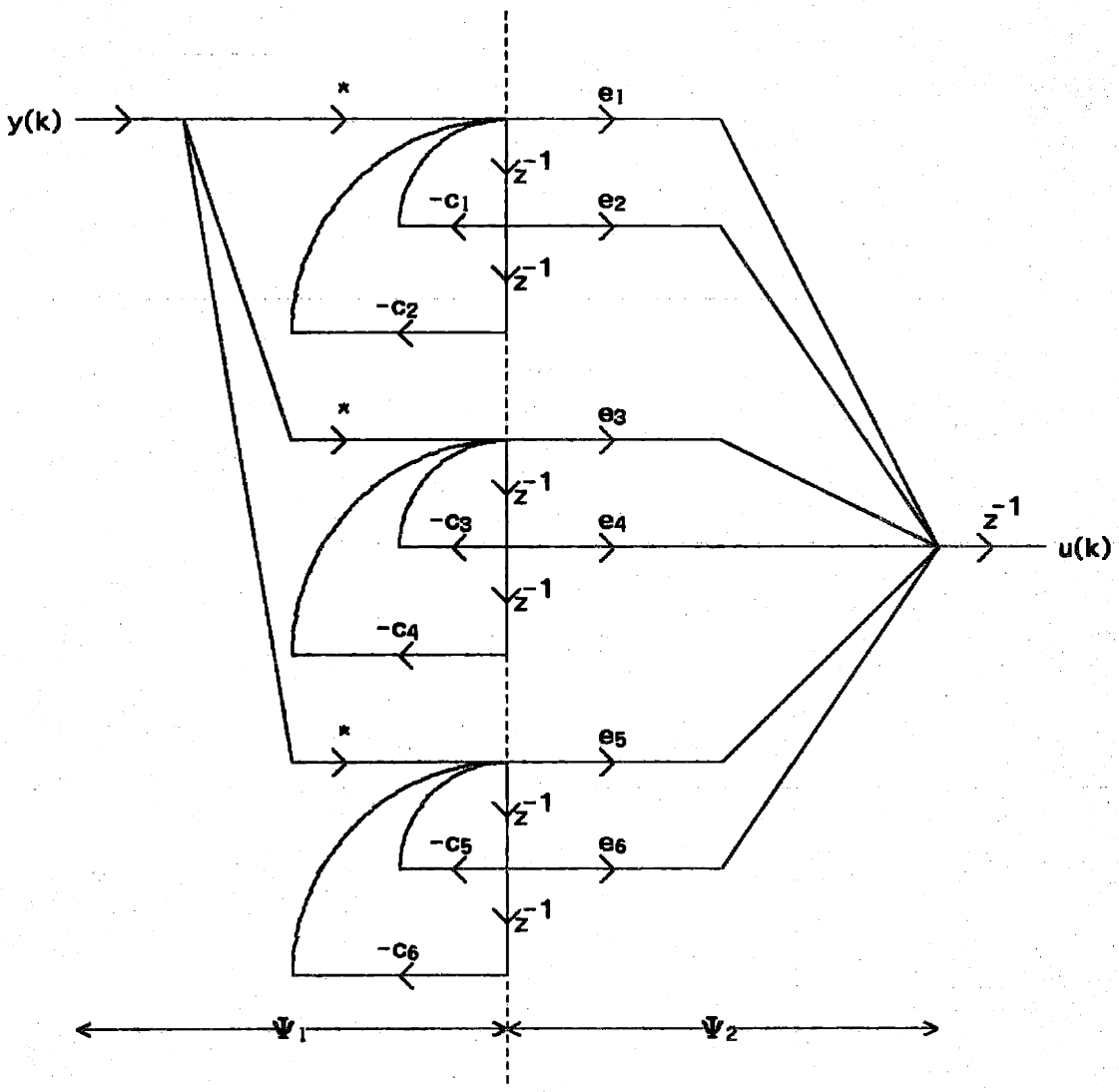



Figure 3-8: Parallel Structure (Direct Form II)

problem arises when we seek to directly implement the transfer function (3.1) with the parameters (coefficients) of equations (2.14):

$$\Psi_3\Psi_2\Psi_1 = \left[\begin{array}{c} I_6 \\ \hline -G \end{array}\right] \left[\begin{array}{c|c|c} \Phi & \Gamma & K \end{array}\right] \left[\begin{array}{c|c} I_6 & 0 \\ \hline 0 & I_2 \\ \hline -L & \end{array}\right] \quad (3.26)$$

where $I_6$ represents a 6×6 identity matrix. In general this structure (termed the *simple* form) has three precedence levels, is canonic in delays, and has up to $n(n+4)$ coefficients, depending on the entries in $\Phi$, $\Gamma$, $L$, $K$, and $G$. For a sixth-order LQG system, this structure would have up to 60 coefficients. This number of multiplies is quite excessive, compared to any commonly-used filter structure. However, this compensator structure (or the similar structure based on the $\Psi_\infty$ of the simple form) is often used for steady-state LQG control applications, more or less by default.

Another broad class of structures includes all the structures whose modified state space representations have just one precedence level matrix. These structures could be called *state space* structures, since the arithmetic and quantization operations involved can be described using state space notation. Some of these can be generated from the direct form II, cascade, parallel, and simple forms just by multiplying the various $\Psi_i$ matrices together to produce $\Psi_\infty$, and using the result as a structure. The standard observable, standard controllable, and Jordan forms [36] well-known to the control and estimation field also correspond to simple one-level structures [15,30]. One could envision such structures being useful for two reasons. First, their performance may be superior to certain multiple-level structures, whether or not they have more coefficients. Secondly, a one-precedence-level structure allows a faster system sampling rate

than a multiple-level structure (see Chapter 4), and thus potentially better performance. An interesting type of one-level filter structure is the *minimum roundoff noise* structure of Mullis and Roberts [18,37,38], and Hwang [39]. Given no constraints on the coefficients of a one-level delay-canonic filter structure, they have derived a technique for computing the coefficient values producing minimum roundoff noise at the filter output. Unfortunately, this filter structure requires $(n+1)^2$ coefficients. To avoid this problem, the authors have also presented *block optimal* filter structures, which are cascade or parallel forms composed of minimum noise second-order sections. (See also Jackson, Lindgren, and Kim [40]). For a block optimal structure, only $4n+1$ coefficients are required. One of the efforts of Chapter 5 will be to extend the ideas of Mullis and Roberts to derive minimum roundoff noise *compensator* structures.

Using $f_1, \cdots, f_m$ as the coefficients, a sixth-order block optimal parallel compensator structure would have the following modified state space representation:

$$\Psi_1 = \begin{bmatrix} f_1 & f_2 & 0 & 0 & 0 & 0 & 0 & f_3 \\ f_4 & f_5 & 0 & 0 & 0 & 0 & 0 & f_6 \\ 0 & 0 & f_7 & f_8 & 0 & 0 & 0 & f_9 \\ 0 & 0 & f_{10} & f_{11} & 0 & 0 & 0 & f_{12} \\ 0 & 0 & 0 & 0 & f_{13} & f_{14} & 0 & f_{15} \\ 0 & 0 & 0 & 0 & f_{16} & f_{17} & 0 & f_{18} \\ f_{19} & f_{20} & f_{21} & f_{22} & f_{23} & f_{24} & 0 & f_{25} \end{bmatrix} \tag{3.27}$$

Note that the pole-zero pairing issue must still be addressed, as with any parallel form. No additional scaling multipliers are required in (3.27). As with any cascade, a block optimal cascade compensator structure would have the disadvan-

tage of having multiple precedence levels; $n_s$ in this case. (Recall that the parallel block optimal structure requires only one precedence level.)

Besides the direct form and general state space forms, there exist other filter structures not derived from a factorization of the transfer function (3.2). Gray and Markel [41] have presented several ladder and lattice forms that are delay-canonic. Another set of ladder filters [42], also delay-canonic, result from *continued-fraction* expansions of (3.2). A ladder structure that has received a great deal of attention in the filtering literature is the wave digital filter [43,44,45]. This filter structure is based on analog LC ladder filters, and directly results from a consideration of the transmission-line equations of microwave filters. Line delay and the transmitted and reflected voltage waves become the sample delay $T$ and the signal variables of the wave digital filter. Characteristics of this structure that derive from the passivity and losslessness of its analog counterpart [46], and lead to the absence of limit cycles under specific sign-magnitude truncation arithmetic. (See Chapter 5). The coefficient sensitivity of this structure has been shown to be comparatively low [44], and under certain additional constraints [48] it will also be low-noise. Additional improvements have been introduced to reduce the number of multiplies [49] and the number of delays [50]. Meerkötter and Wegener [51] have developed a second-order wave digital filter section which can be the building block of a cascade or parallel form. This section would have four multiplies and two sign-magnitude truncation quantizers, but require five additional scalers (as opposed to the one or two scalers of most sections). As with many of the digital filter structures, ladder-type structures could easily be adapted for compensator structures by adding a series delay to the filter structure output.

Finally, a general class of *optimal* structures exists. Chan [17] has described a technique for filters where, through the use of the transformations in (3.15) and (3.16), a scalar function of the structure parameters can be minimized. More importantly, the method will hold almost any set of $\Psi_i$ entries constant, as desired. Thus we can control the number of coefficients in the structure and their locations while minimizing roundoff noise or coefficient quantization effects, or some combination of the two. Chapter 8 will adapt this useful technique for the optimization of compensator structures, and an example of the constrained minimization of compensator roundoff noise effects will be presented.

This discussion of compensator structures was not intended to present an exhaustive list of possible structures, but only a representative selection. (For example, *transpose* configurations [30,31] were not considered.) The analyses in Chapters 5, 6, and 7 compare some of these compensator structures with respect to their finite wordlength properties. The overall aim is to provide the reader with a basic grasp of the various structures and of the different criteria for choosing among the different classes of structures, given control and estimation applications.

## §3.4 Summary

Beyond a presentation of the more common types of compensator structures, the main point of this chapter was the introduction of the modified state space representation. This representation exactly reflects the computations that determine the performance of a compensator structure when implemented with finite wordlengths, and also the order in which these computations must occur. This representation, unlike the form introduced by Chan [17] which is adequate

for digital filters, must include all the inherent delays necessary to complete the operations within the compensator structure. Finally, as with the Chan form, it is possible to apply simple transformations to this representation in order to synthesize a compensator structure with superior finite wordlength performance.

# Chapter 4: Architectural Issues: Serialism, Parallelism, and Pipelining

## §4.1 Introduction

In this chapter, we will examine the architectural issues involved in the implementation of digital feedback compensators. We will show that the basic concepts of serialism and parallelism as they apply to digital filter structures represented in Chan's notation extend without modification to digital compensator structures represented in the modified state space notation. However, the same cannot be said concerning the application of pipelining techniques to compensators. In fact, we will show that pipelining in control systems brings out another important issue: the interaction between the ideal design procedure described in Chapter 2 and the implementation of the resulting compensator.

Perhaps the most basic issue in any consideration of digital system architecture involves the concepts of serialism and parallelism [31,52,53]. Essentially, this notion involves the degree to which processes, or operations, in the system run in sequence (serially) and the degree to which they execute concurrently (in parallel). At one extreme, *any* system can be implemented with a completely serial architecture, executing all its processes one at a time. This procedure requires the minimum number of actual hardware modules and the maximum amount of processing time for completion of the system task. On the other hand, any system can also be implemented with a maximally-parallel architecture, having as many concurrent processes as possible. Such a design requires the maximal amount of hardware, but completes the overall system task in minimum time. Thus, the serialism/parallelism tradeoff is another example of the frequently encountered space-time tradeoff [52].

There is an important asymmetry implicit in the exploitation of serialism and parallelism. It is always possible to execute processes one at a time (totally serially). However it is *not* always possible to execute them all at once (in a totally parallel manner). There is a minimum amount of serialism required. Figure 4-1 gives a typical example, consisting of three processes (P1, P2, and P3), and

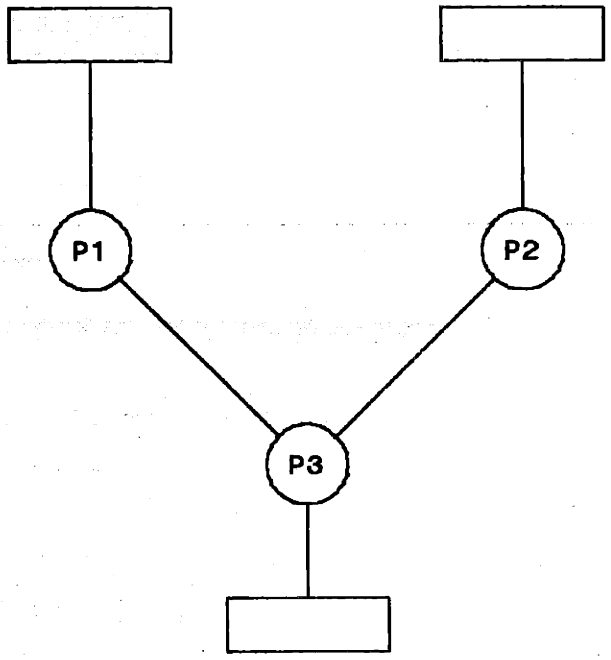



Figure 4-1: Three-Process System

data cells [52] for input and output. Assume that each of the three processes require $t$ seconds for completion (given specific hardware modules) and that each process executes as soon as all of its inputs are valid. Given a general-purpose computing module, then clearly a serial architecture that would require $3t$ seconds to complete the overall task is possible. On the other hand, figure 4-1 clearly shows that processes $P1$ and $P2$ must be finished *before* process $P3$ can begin.

Consequently, only processes $P1$ and $P2$ can operate in parallel. For such an architecture, two hardware modules would be required, and the total computation time would be reduced to $2t$ seconds. The totally-parallel architecture (total time $t$ with three hardware modules) is not possible for the system of figure 4-1.

Under certain conditions, this 'speed barrier' can be broken through the use of pipelining [31,62]. If the original objective of the system is to perform a task *repeatedly* (as soon as the present task is completed, a new task begins), then pipelining could realize an effective throughput rate equal to (or at least closer to) that of a totally-parallel architecture. Reconsider figure 4-1. Suppose that a separate hardware module is reserved for each process, the sampling rate is $\frac{1}{2t}$, and the maximally-parallel $2t$ second architecture is used. The input and output data cells now represent registers clocked at rate $\frac{1}{2t}$. Let us examine any $2t$-second interval. During the first $t$ seconds, module 3 (for executing process $P3$) will be idle, since its inputs are not yet valid. During the last $t$ seconds, module 3 will be active and modules 1 and 2 will be idle. The total $2t$ second time from a task initiation until its completion cannot be reduced without faster hardware modules. However, the idle modules can be put to use by pipelining the processes. While module 3 is active and modules 1 and 2 otherwise idle, the next task may as well begin and use modules 1 and 2. The net result (in this example) is a doubling of the throughput rate (task completions per second) from $\frac{1}{2t}$ to $\frac{1}{t}$. It must be stressed here that any given task still takes $2t$ seconds from start to finish; however, successive task completions occur at $t$ second intervals. In terms of hardware required, the pipeline would be effected by adding two

clocked registers to buffer the intermediate results from modules 1 and 2, and of course by doubling the clock rate. Figure 4-2 shows two ways of viewing the

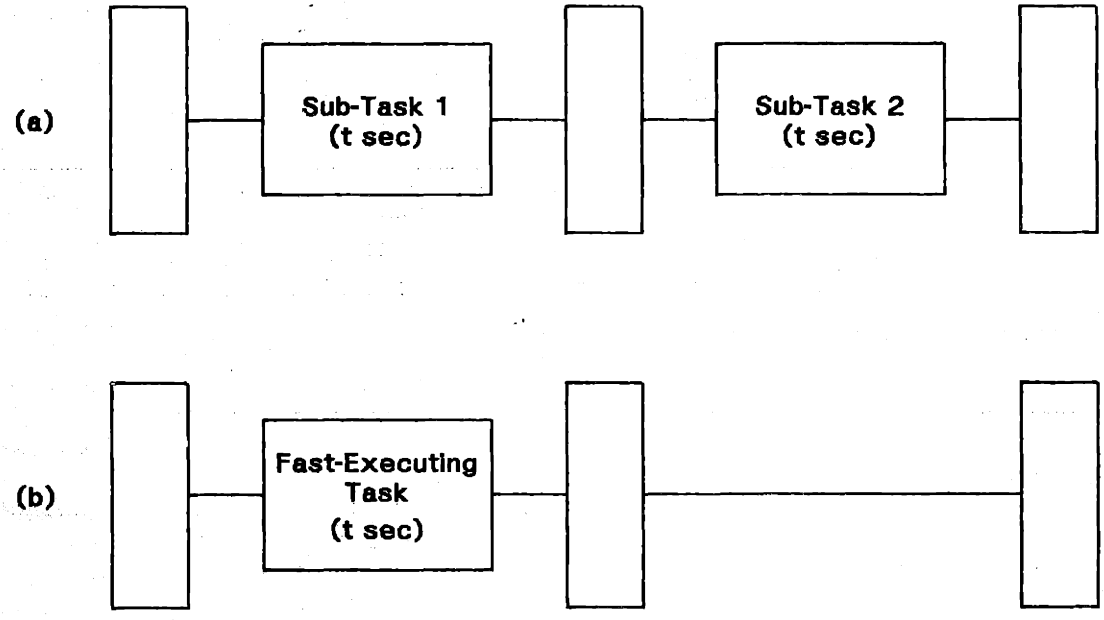



Figure 4-2: Models For Pipelining

pipelined case for this example. Basically, the pipeline splits a larger task not implementable in a totally-parallel architecture into smaller sequential sub-tasks, each of which can be implemented in a totally parallel fashion (figure 4-2a). An equivalent viewpoint (figure 4-2b) considers pipelining to be represented by a faster-executing task coupled with some serious delay (inherent in the additional clocked registers).

An important application of pipelining is in the implementation of digital filter structures [31,54]. In such a case, the system task corresponds to the generation of a filtered output value from an input sample, and the individual processes correspond to the hardware digital multiplications and additions that exist in the

particular structure implemented (ignore A/D and D/A operations for now). Figure 4-3a shows a two-pole digital filter with input *y* and output *u*. As shown, the unit

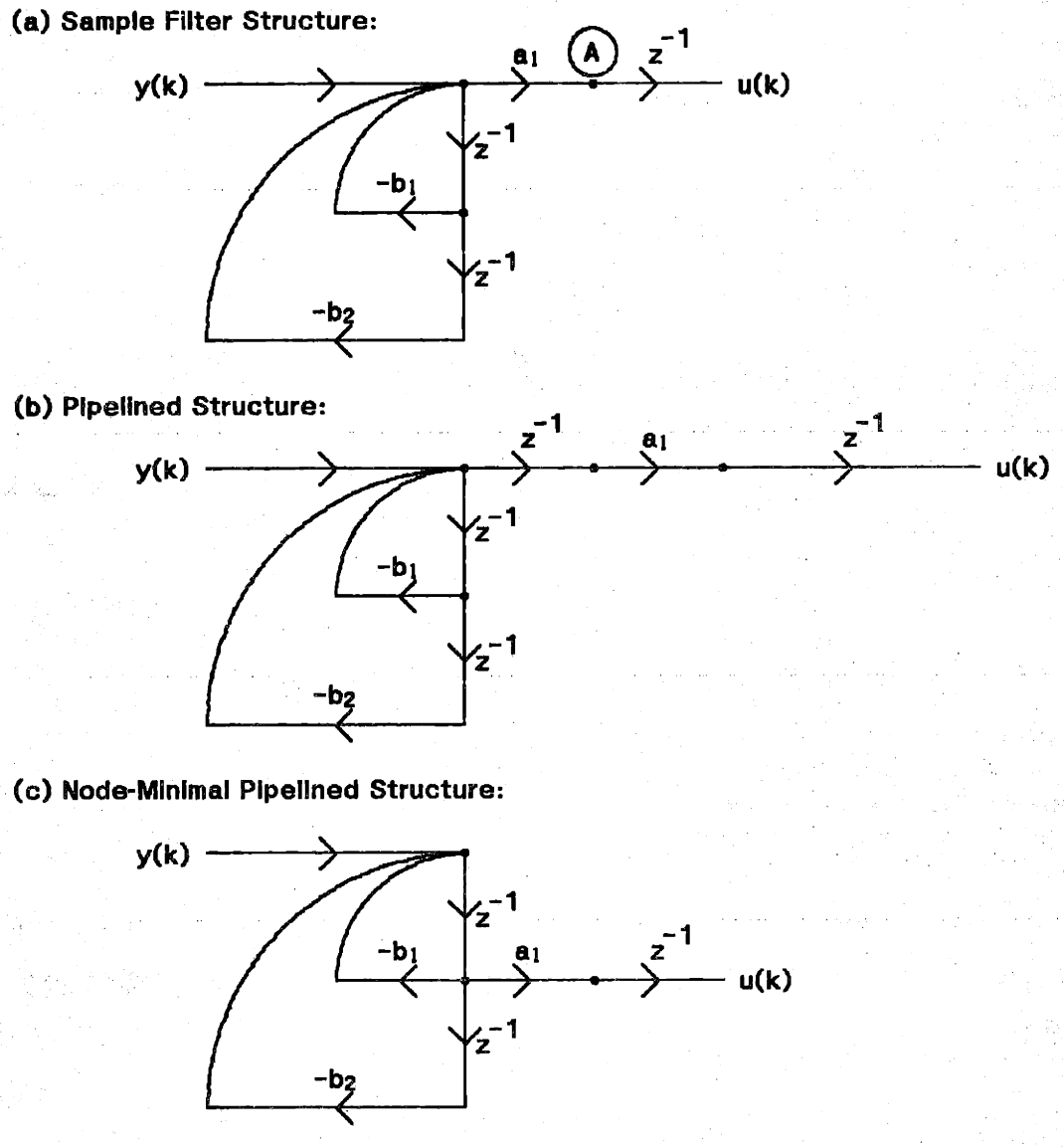



Figure 4-3: Pipelining a Simple Digital Filter

delay $z^{-1}$ represents a clocked storage register. Thus, all the arithmetic and quantization operations have one sampling period in which to be completed. Com-

puting the signal $u(k+1)$ at node $A$ in figure 4-3a requires three multiplications and an addition. The multiplications involving $b_1$ and $b_2$ can operate in parallel, then the addition occurs, and finally the multiplication by $a_1$. Using three hardware multipliers instead of two, and assuming negligible add time, the multiply operations can be pipelined and the sampling rate doubled. The new configuration could be implemented with just one additional storage register, represented in figure 4-3b as an additional unit delay. However, this new signal flow graph is not *node minimal*, since it contains two states that are *exactly* equivalent. Removal of one of these states produces the node-minimal signal flow graph shown in figure 4-3c. Thus, the pipelined structure of figure 4-3c has the same number of unit delays (storage registers) as the original structure in figure 4-3a. For this particular example, pipelining did not require the use of more unit delays. This would not be true in general. Note that each $z^{-1}$ in figures 4-3b and 4-3c represents only half the delay time of those in figure 4-3a if the sampling rate is doubled, as made possible by pipelining.

From the example of figure 4-3, it is clear that pipelining ties in closely with the digital filter notion of precedence. Specifically, let us consider *node precedence*, that is, the precedence relations involved in the addition, multiplication, and quantization operations needed to compute the node signals. In this case, the modified state space representation (See Chapter 3) is very convenient since it explicitly shows the number of precedence levels involved. If a structure represented in this notation has only one precedence level, then it can have a totally-parallel architecture (parallel in terms of the multiply/add computations involved in each precedence level). If more than one such level is required, no totally-parallel architecture is possible, and the number of levels $q$ will equal the

minimum degree of serialism required. Pipelining, if applicable, would actually *change* the structure by inserting unit delays so that a new structure (one with fewer levels and thus a faster sample clock rate) is formed. The pipelined structure would have the same transfer function as the original non-pipelined structure, except for some series delay, and would probably have more state nodes. Series delay is of little consequence in most digital filtering applications. Thus a two-level structure can be designed for a sampling period of $\frac{t}{2}$ even though the calculations require $t$ seconds, since pipelining (given a two-level structure) will fit the calculations into a $\frac{t}{2}$ slot at the expense only of a series delay of $\frac{t}{2}$ seconds. Equations (4.1) through (4.4) show the modified state space representations and transfer functions of the non-pipelined (sampling period $t$) and pipelined (sampling period $\frac{t}{2}$) filters of figure 4-3a and 4-3c respectively:

$$\Psi_2\Psi_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & a_1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -b_2 & -b_1 & 0 & 1 \end{bmatrix} \tag{4.1}$$

$$H_{np}(z) = \frac{a_1 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{4.2}$$

$$\Psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -b_2 & -b_1 & 0 & 1 \\ 0 & a_1 & 0 & 0 \end{bmatrix} \tag{4.3}$$

$$H_p(z) = \frac{a_1 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \qquad (4.4)$$

Note the reduction from two levels to one level (see (4.1) and (4.3)), allowing the doubled sampling rate, and also the extra $z^{-1}$ factor in the numerator of (4.4). The number of states in (4.3) remained at three since no additional storage registers were actually added to effect the pipeline.

Let us now consider pipelining as it applies just to the multiply operations in a structure. Such a consideration will be valuable whenever the multiply time dominates over all the addition and quantization operation times in a structure, a situation that is not uncommon in microprocessor-based digital systems. Since we are neglecting all calculation times other than the multiply times, it is sufficient to know the precedence to the multiply operations *alone* in order to determine the architectures that are possible. Thus the node precedence evident from the different $\Psi_i$ matrices of a modified state space representation will not be adequate to describe the *multiplier precedence* relations. Such relations can be determined from the signal flow graph or from an examination of the specific location of each multiplier coefficient in the $\Psi_i$ matrices. In either case, the multipliers can be grouped into precedence *classes*. Frequently, the number of *multiplier precedence classes* and *node precedence levels* will be the same, but the multiplier coefficients in class 1 (of highest multiplier precedence) and the multiplier coefficients in node precedence level 1 (the matrix $\Psi_1$) need not be identical. It *will* be true that all the multiplier coefficients in the matrix $\Psi_1$ will also be in multiplier precedence class 1. Furthermore, multiple-level structures often have fewer multiplier classes than node precedence levels.

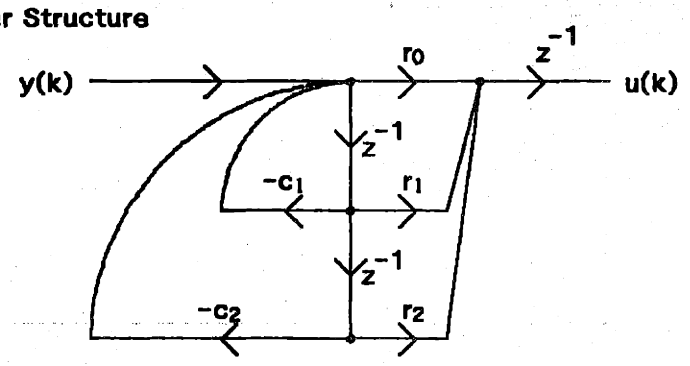
As an example, consider the cascade structure of figure 3-6 and its modified state space representation (3.22). Assume all scaling multipliers to be simple shifts (powers of two); thus they are not considered to be true coefficients requiring hardware multipliers. All the multiplications of coefficients by state node or input signals can occur immediately after each sampling instant and therefore fall in multiplier precedence class 1. Thus the $c_1$, $c_2$, $c_3$, $c_4$, $c_5$, $c_6$, $d_2$, $d_3$, $d_4$, $d_5$, and $d_6$ multiplies can operate in parallel given enough hardware multiplier modules. Only the $d_1$ multiplication lies in class 2; it must await the completion of the $c_1$ and $c_2$ multiplies. Of course, given the two classes and 12 multiplies, an optimal, that is maximal, use of the hardware is made with only 6 hardware multipliers (assuming no pipelining). Five of the class 1 multiplies (but not $c_1$ or $c_2$) would be computed in the *second* multiply cycle with the $d_1$ multiply. Thus the cascade of figure 3-6 has two multiplier precedence classes, although it has four node precedence levels. Similarly, the cascade structure in figure 3-7 has only one multiplier precedence class assuming power-of-two scalers, although its modified state space representation (3.23) shows three node precedence levels. If in fact general scalers are used in these two cascades, they will constitute multiplier coefficients, and the number of multiplier precedence classes and node precedence levels will be the same. No matter what type of scalers are used, the parallel structure of figure 3-8 has the same number of multiplier classes as it has node precedence levels; even so, the coefficients of multiplier class 1 ($c_1$, $c_2$, $c_3$, $c_4$, $c_5$, $c_6$, $e_2$, $e_4$, and $e_6$) are *not* simply the coefficients in $\Psi_1$. The coefficients $e_1$, $e_3$, and $e_5$ belong to multiplier class 2 because they must await the completion of the $c_1$ through $c_6$ multiplies. This no-

tion of multiplier precedence is more completely formulated in [31], but the basic conclusion is as follows: although the modified state space representation correctly describes the operations that must occur in computing the node values within a structure and has other useful properties (see Chapter 3), the multiplier precedence relations (more easily seen directly from the signal flow graph) are more significant for determining the possible hardware architectures when the multiply time is dominant.

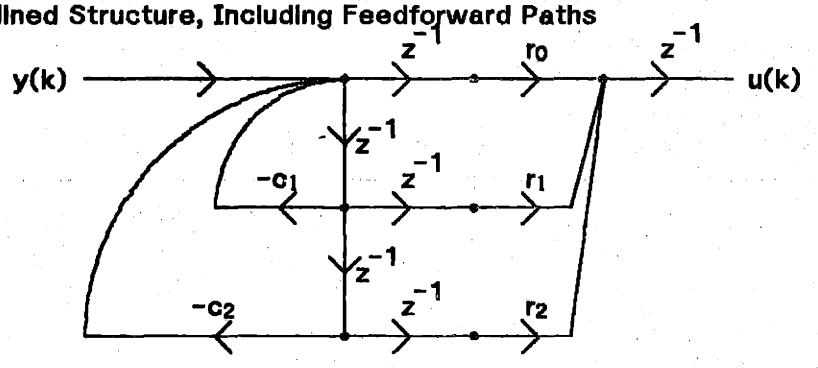
## §4.2 Restrictions on Pipelining

Certain basic restrictions [31] must be observed when pipelining a complex structure. The first limitation in applying pipelining concerns parallel data paths within the structure. Whenever any portion of a system is pipelined to increase the sampling rate (which adds effective delay), all parts of the system that feedforward in parallel with the pipelined portion must receive equivalent actual delay in order to maintain the desired transfer function. In other words, the data flowing through the system must remain synchronized whether or not pipelining is applied. Consider the second-order digital filter of figure 4-4a. A direct pipelining of this structure by adding a unit delay preceding the $r_0$ multiplier, as done with figure 4-3a, will result in a very different transfer function than the original one. To preserve the transfer function desired, except for series delay, unit delays must also be inserted in the parallel feedforward branches $r_1$ and $r_2$. This new (one-level) structure appears in figure 4-4b but is not node-minimal. Figure 4-4c shows an equivalent node-minimal structure, requiring only one additional state instead of three. Its modified state space representation is shown in (4.6):

(a) Filter Structure




(b) Pipelined Structure, Including Feedforward Paths




(c) Node-Minimal Pipelined Structure

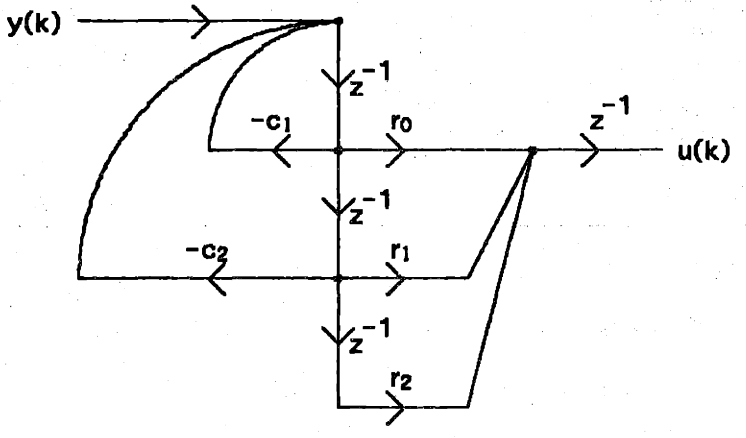



Figure 4-4: Pipelining and Feedforward Data Paths

$$\Psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -c_2 & -c_1 & 0 & 1 \\ r_2 & r_1 & r_0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

The second difficulty encountered in applying pipelining techniques involves feedback. Suppose there exists a series of operations which makes up part of a closed feedback loop within a structure. Pipelining these operations would result (as with the previous example) in a very different transfer function. Consider the filter of figure 2-5. Its transfer function and two-level modified state space representation are shown in equations (4.6) and (4.7):

$$H(z) = \frac{r_0 z^{-1}}{1 + (c_1 - r_0) z^{-1}} \tag{4.6}$$

$$\Psi_2 \Psi_1 = \begin{bmatrix} 1 \\ r_0 \end{bmatrix} \begin{bmatrix} -c_1 & 1 & 1 \end{bmatrix} \tag{4.7}$$

If we pipeline by inserting a delay preceding $r_0$ (or by equivalently moving the $r_0$ branch to state node $v_1$), the modified state representation will indeed show only one level:

$$\Psi_1 = \begin{bmatrix} -c_1 & 1 & 1 \\ r_0 & 0 & 0 \end{bmatrix} \tag{4.8}$$

However, the overall transfer function is now quite different:

$$H(z) = \frac{r_0 z^{-2}}{1 + c_1 z^{-1} - r_0 z^{-2}} \qquad (4.9)$$

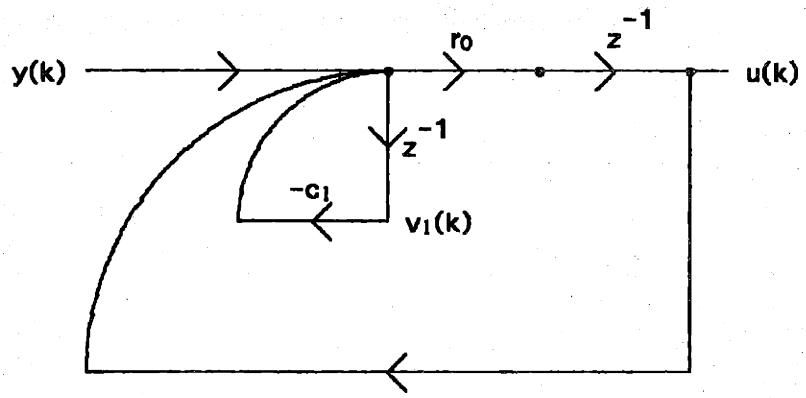



Figure 4-5: Filter with Output Feedback

Although part of the feedback loop has been 'sped up' by pipelining, the delay introduced prevents the feedback term from being equivalently sped up. (The data is not synchronized.) Thus, pipelining within a feedback loop is ordinarily avoided.

## §4.3 Pipelining Feedback Compensators

In the context of the control problem formulated in Chapter 2, the ideas of serialism and parallelism apply unchanged to the implementation of digital controller architectures. However, since a global feedback loop exists around the entire compensator, that is, through the plant, pipelining seems to be out of the question, as shown in the example of figure 4-5. Suppose that we design an LQG compensator for a system with a sampling rate of $\frac{2}{T}$, the resulting compensator has two multiplier precedence levels, and the multiply time $t_m$ equals $\frac{T}{2}$. Pipelining

would seem to be necessary unless we were willing to drop the sampling rate to $\frac{1}{T}$. Unfortunately, the series delay that would result from pipelining this compensator would introduce an unplanned-for pure time delay. The deleterious effects of pure time delay (linearly-increasing negative phase shift) on the stability and phase margin of a feedback system are well known. Even if instability does not result, the performance index $J$ will be larger than expected and the qualitative dynamic performance will be compromised.

Fortunately, there is an approach to pipelining that will be effective for control systems. Consider the LQG system and compensator design technique described in Chapter 2. Assume that for some original controller design, the sampling interval is not long enough to complete all the calculations involved in the compensator (which is the situation as described above). In principle, pipelining techniques could help, but unavoidable delay would be introduced. An effective use of pipelining simply means that we somehow include this unavoidable delay in the original design procedure. This aim can be realized through *state augmentation* [1]. Suppose that pipelining would allow a factor of two increase in the sampling rate, thus adding only a single series delay. If the plant is described at the doubled sampling rate $\frac{2}{T}$ by (4.10):

$$\begin{aligned} x(k+1) &= \Phi\, x(k) + \Gamma\, u(k) + w_1(k) \\ y(k) &= L\, x(k) + w_2(k) \end{aligned} \tag{4.10}$$

(recall that the matrix parameters above depend on $T$) then, preceding $u(k)$ with the series delay to form $\tilde{u}(k)$, the augmented plant can be modelled as follows (see figure 4-6):

$$\begin{aligned} \tilde{x}(k+1) &= \begin{bmatrix} \Phi & \Gamma \\ 0 & 0 \end{bmatrix} \tilde{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tilde{u}(k) + \begin{bmatrix} w_1(k) \\ 0 \end{bmatrix} \\ y(k) &= \begin{bmatrix} L & 0 \end{bmatrix} \tilde{x}(k) + w_2(k) \end{aligned} \tag{4.11}$$

where $\tilde{x}(k+1) = \begin{bmatrix} x(k+1) \\ u(k+1) \end{bmatrix}$. For this augmented system, the weighting matrices $Q$ and $M$ in the expression for the performance index (2.6) must also be augmented, adding an all-zero row and column to $Q$, and a single zero element to $M$. The weighting parameter $R$ will be the same as for the system (4.10). Now we must treat (4.11) as a new system and design an LQG compensator for it. Then *that* design can be pipelined, which introduces the inherent added delay shown in figure 4-6.

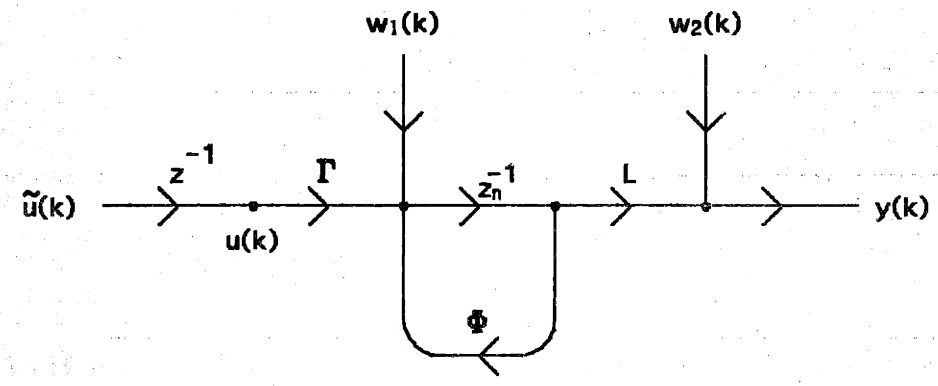



Figure 4-6: State Augmentation for Control System Pipelining

For this situation, two observations can be made. First, the Kalman filter portion of the LQG design for (4.11) will have what seems to be a difficulty due to the added delay — the numerical routines blow up. Common sense dictates however that there is no need to estimate $\tilde{x}_{n+1}(k) = u(k)$ since it is the *actual* plant input, which is known. Thus we need only estimate $\tilde{x}_1(k)$ through $\tilde{x}_n(k)$, namely

the vector $x(k)$. That estimation problem has already been solved as the $n^{th}$-order Kalman filter for (4.10), with gains $k_1$ through $k_n$. Using these results, the optimal filtering gains for the augmented system (4.11) can be written:

$$\tilde{k} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \\ 0 \end{bmatrix} \tag{4.12}$$

The $(n+1)^{th}$-order optimal regulator problem for (4.11) can be solved with no difficulty at all.

The second observation that we can make for this augmented-system pipelining technique involves the *consistency* of the design technique. A delay-canonic structure for the optimal LQG compensator for (4.11) will be of order $n+2$ since (4.11) is of order $n+1$, and *not* of order $n+1$ as is the canonic compensator structure for (4.10). Thus this approach to controller pipelining gives rise to a compensator of higher dimension (more poles), requiring more states (delay elements) and more coefficients. Along with this increase in order comes a more important point — the new higher dimensional compensator structure must allow the same degree of pipelining as the original structure, or the whole controller pipelining design procedure is invalid, that is, inconsistent. This point is especially of concern when using structures whose number of precedence levels is a function of the number of compensator states (for example, the cascade forms). As an example, consider a second-order plant and a direct form II compensator structure, which requires three delays and two precedence levels. To exploit pipelining, we must augment the plant and redesign the compensator — its direct form II structure now re-

quires four delays (states). There would still be only two (node or multiplier) precedence levels as before, so pipelining to double the sampling rate will work as planned. However, if we decide to use a *cascade* of two direct form sections (assume one second-order section, one first-order section, and general non-power-of-two scaling multipliers), then the result is three precedence levels. Pipelining to allow the $\frac{2}{T}$ sampling rate will *not* now result in the effect of a single added unit delay as assumed, but will involve *two* series unit delays, making the design procedure invalid. In other words, if we implemented the pipeline as described above, the system would not perform as expected; more delay would be present in the loop than had been accounted for in the design. Such problems can be avoided with a proper choice of structure.

There is one positive note associated with the increased dimensionality of the compensator, and it is related to the particular form of (4.12). Usually, an increase in dimension (number of states) by one involves at least two additional coefficient multipliers. (A fifth-order plant requires a compensator with at least 10 coefficients, a sixth-order plant requires one with 12 coefficients, etcetera — see figure 3-5) However, by virtue of the zero entry in (4.12), the general form of the compensator transfer function for the augmented system is simpler:

$$H(z) = \frac{a_2 z^{-2} + a_3 z^{-3} + \cdots + a_{n+1} z^{-(n+1)}}{1 + b_1 z^{-1} + \cdots + b_{n+1} z^{-(n+1)}} \qquad (4.13)$$

Comparing (4.13) to (3.2) shows a difference of only one coefficient — not two. This fact helps make the pipelining approach a bit more attractive, at least with certain structures (for example, any direct form and any cascade or parallel structure based on a direct form.)

One last general point should be mentioned. The application of any pipelining technique or the use of parallelism to increase the sampling rate is desirable only if it allows a decrease in the performance index $J$, or in whatever gauge of system performance one accepts. However, not all systems have a performance measure that decreases (improves) monotonically with decreasing $T$ [25]. Intuitively, any system with sharp resonances will lose controllability (implying a large $J$) when the sampling frequency is near a resonance. One must be aware of such cases. If such a case does not occur, then pipelining will reduce the performance index, although certainly not as much as the (non-implementable) straightforward rate-$\frac{2}{T}$ LQG compensator design which adds no delay. Whether this pipelining approach is effective *enough* to warrant the higher-order compensator depends on the designer's particular application.

## §4.4 Controller I/O Pipelining

One common application of pipelining in a feedback environment involves the often time-consuming compensator input/output (I/O) operations, namely, the sampling and the A/D and D/A conversion operations. Let us assume that a structure with one multiplier precedence level (for example, the block optimal parallel structure of (3.27)) is chosen to implement a compensator, and that a totally-parallel architecture is used for the multipliers involved. The compensator can then be modelled as a three-process task (figure 4-7). With no pipelining the minimum sampling period $T$ equals $t_1+t_2+t_3$ seconds. Assume that the slowest process is the multiply time and that $t_2 = t_1+t_3 = \frac{T}{2}$. If we now pipeline these three processes, a factor of two increase in throughput and sampling rate is pos-

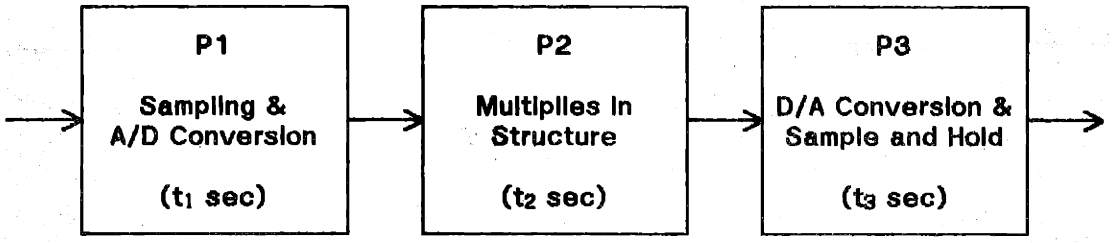



Figure 4-7: Three-Process Compensator Model

sible. (Throughput rate is limited by the slowest process). At each sample time, sampling and A/D conversion of a new $y$ sample would begin. Then $t_1$ seconds later the structure multiplications could begin, overlapping the next sampling and A/D operation. Figure 4-8 diagrams the processes occurring in such an I/O

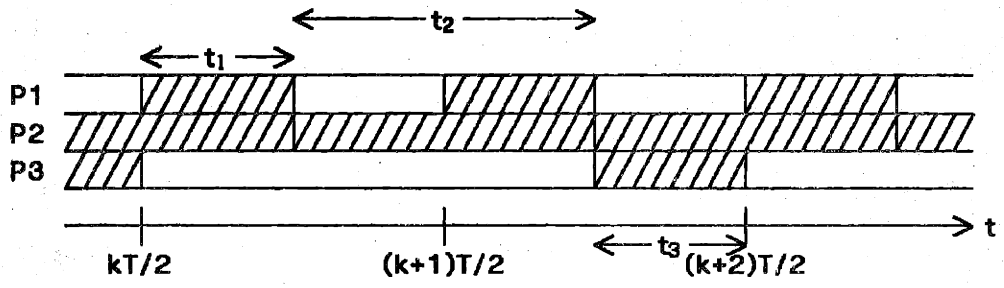



Figure 4-8: Concurrency of Processes in I/O Pipelined Compensator

pipelined compensator with increased sampling rate $\frac{2}{T}$. Note that the hardware multipliers will now be active 100% of the time.) We can represent this pipelined system as the designed compensator structure followed by a series unit delay resulting from the pipeline. Since part of this unit delay is involved in buffering the intermediate A/D results and the rest is involved in buffering the multiplier

results from the structure, two hardware storage registers will be required for this example. However, their clock signals will be staggered, since the three operations of figure 4-8 take different amounts of time to complete. Basically, these clock signals (all of period $2/T$) must be phased so that the results from each process are stored as soon as they are completed. Thus register 1 is clocked by sample pulses delayed by $t_1$ seconds, and register 2 is clocked by sample pulses delayed by $t_1+t_2$ seconds. (This phasing is shown as fractional delay time in the simple example of figure 4-9.)

If we apply the design technique outlined in section 4.3 to produce a (pipelineable) compensator for this I/O case, the order of the compensator will of course be one greater than the non-pipelined design, implying at least one additional state and coefficient. No matter what the plant dimension may be, a block optimal parallel structure (or any state-space structure — see section 3.3) will have only one precedence level. Thus, I/O pipelining with a one-level compensator structure results in a valid design procedure.

## §4.5 Compensator I/O Pipelining Examples

Four examples have been selected to illustrate what can occur with compensator (I/O) pipelining. Each example consists of four cases. Case 1 represents the plant discretized at a $T$ second sampling period with its corresponding LQG compensator (no pipeline). Case 2 represents the plant discretized at a $\frac{T}{2}$ second sampling period with *its* corresponding LQG compensator. This case does not include any pipelining, but is not physically implementable due to the short sampling interval. The performance index for this case consti-

tutes an unreachable lower bound to the performance of the augmented-plant approach to pipelining (case 3)). Case 4 (blind pipelining) results when the compensator designed for case 2 is pipelined in order to make it physically implementable. Thus the delay due to the pipeline is *ignored* in the pipelined design, usually resulting in a performance level that is worse than the non-pipelined level (and perhaps even in a system that is unstable). Assuming that $J$ is a monotonic increasing function of $T$, we can expect that the different cases will rank, from highest $J$ to the lowest, as follows: case 4, case 1, case 3, case 2. (It is possible but unlikely that case 4 could have a lower $J$ value than case 1.) Remember, however, that case 2 is not implementable.

The simplest I/O pipelining example consists of a single-input, single-output, single-integrator plant:

$$\begin{aligned} \dot{x}[t] &= u[t] + w_1[t] \\ y[t] &= x[t] + w_2[t] \end{aligned} \tag{4.14}$$

where $T$=6 seconds. Referring to Chapter 2, equations (2.1)-(2.3), the parameters $\hat{Q}$ and $\hat{R}$ were both chosen to be 1 and the noise intensities $\Xi_1$ and $\Xi_2$ were selected to be 0.3 and 0.125. Figure 4-9 illustrates the discretized system and the form of the compensator before pipelining (case 1) and after pipelining through state augmentation and redesign (case 3). A one-level version of the direct form II structure (obtained from the $\Psi_\infty$ matrix of the direct form II, as mentioned in section 3.3) is used for the compensator. Note the inclusion of the two fractional delays (registers) in figure 4-9b, as mentioned earlier in this section. The form of the system for case 2 would look the same as that in figure 4-9a; however the gains of all the branches would differ. For case 4, we need only

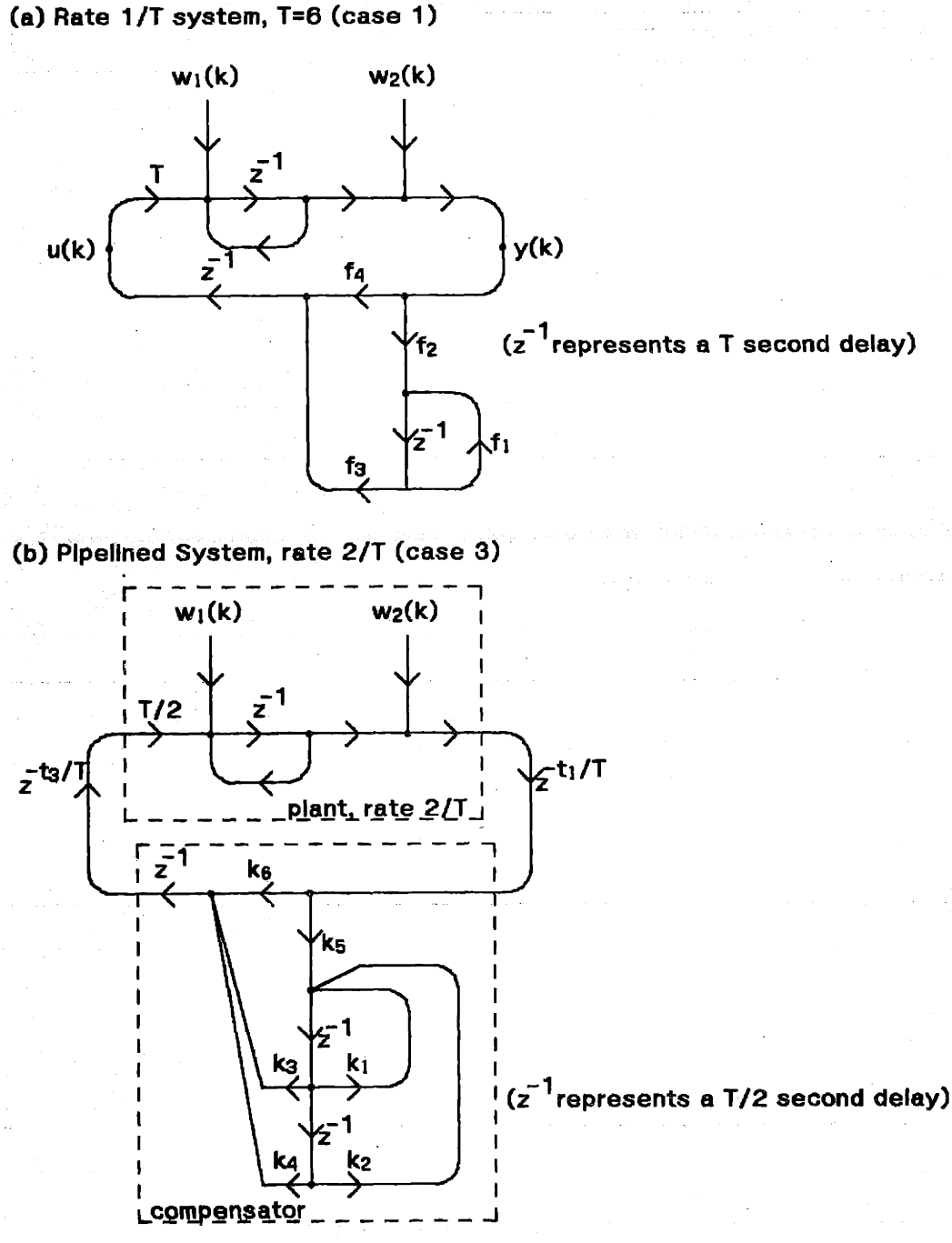



Figure 4-9: Compensator I/O Pipelining for the Single-Integrator Plant

add one series delay to the signal flow graph of case 2.

Three other examples are also considered; a double-integrator plant, a two-state harmonic oscillator plant, and a sixth-order plant derived from the longitudinal dynamics of the F8 fighter aircraft (see Chapter 5 and Appendix A). The continuous-time parameters of the double-integrator system are shown below:

$$\dot{x}[t] = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x[t] + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$y[t] = [1 \quad 0] \tag{4.15}$$

For this system, the continuous-time parameter $\hat{Q}$ was a 2×2 identity matrix, $\hat{R}$ was 1, $\Xi_1$ was the diagonal 2×2 matrix diag(0.2, 0.3), and $\Xi_2$ was 0.125. For the harmonic oscillator, all the parameters were the same as for the double-integrator system, except for the $A$ matrix which is given below:

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{4.16}$$

The performance indices for all the various cases are shown in figure 4-10.

Key:
Case 1 — rate 1/T system
Case 2 — rate 2/T system (not implementable)
Case 3 — rate 2/T pipelined system designed via state augmentation
Case 4 — blind pipelining

| example plant | T | Case 4 | Case 1 | Case 3 | Case 2 |
|---|---|---|---|---|---|
| single integrator | 6 | (unstable) | 2.42 | 2.05 | 1.34 |
| double integrator | 6 | (unstable) | 328 | 179 | 53.2 |
| harmonic oscillator | 6 | (unstable) | 32.7 | 12.9 | 9.72 |
| 6-state F8 plant | 1 | .0038 | .00312 | .00282 | .00222 |

Figure 4-10: *Compensator I/O Pipelining*

Under case 4 we see the consequences of pipelining and ignoring the delay in-

curred. Three of the example systems actually became unstable, and with the fourth, the index $J$ *increased*. As expected, all the case 2 indices were lower than case 1, with case 3 lying between the two. To judge the effectiveness of the state-augmentation pipelining method of case 3, one must examine the degree of improvement in $J$ relative to the possible improvement (the difference between cases 1 and 2). The best improvement shown was for the harmonic oscillator, which is no surprise since the oscillator's natural frequency of $\frac{1}{2\pi}$ radians/second is close to the unpipelined sampling rate $\frac{1}{T}$. The remaining three examples also showed significant improvement. Again, whether or not the pipelineable compensator (with one extra state and at least one extra coefficient) is to be used will depend on the particular level of performance desired and the penalty involved in complicating the hardware.

## §4.6 Summary

To summarize this chapter briefly; section 4.1 introduced the architectural notions of serialism, parallelism, and pipelining, and explained the hardware cost/execution time tradeoff tied to these issues. The issues of serialism and parallelism were shown to involve the same considerations for digital compensators as for digital filters. Section 4.2 discussed the limitations of pipelining techniques, especially the one concerning pipelining in a closed loop (feedback). The extra delay incurred due to the use of pipelining had a deleterious effect on the performance of the feedback system. This problem made the consideration of pipelining for feedback compensators very different than in the case of digital filters. Section 4.3 developed a design technique based on state-augmentation

for dealing with the problem of control system pipelining. Finally, the last section treated a typical application of pipelining techniques to microprocessor-based control systems. For this application, the compensator Input/Output operations and multiply operations could be pipelined to realize a doubling in the system sampling rate. Four examples were presented to illustrate the technique.

# Chapter 5: Finite Wordlength Effects: Quantization Noise

## §5.1 Introduction

One of the major implications of the use of finite wordlengths within a compensator is the necessity of having the nonlinear operations of quantization and overflow in the structure. First, the input A/D unit must convert an analog signal to a fixed-point representation with a specific number of bits $n_{ad}$. (Commercially available units typically produce 6, 8, 10, 12, or 16 bits). This procedure involves an implicit quantization of the input level to one of the set of possible $n_{ad}$-bit words and constitutes an *approximation* (a source of error). The remainder of a structure's quantizers are required by the multiply operations within the structure. Given $n_r$-bit digital words for the node signal variables, then any multiplication by $n_c$-bit coefficients produces an $(n_r+n_c)$-bit product. To store this result in an $n_r$-bit (state) storage register, or to serve as an $n_r$-bit input to another multiplier, requires a quantizing operation. Furthermore, the addition of two $n_r$-bit fixed-point words could produce an extra significant bit, which requires another nonlinear operation to keep the wordlength at $n_r$ bits. Discussion of such *overflow* nonlinearities will be deferred to Chapter 7.

The A/D and multiplier quantizations mentioned above introduce two types of undesirable effects, classifiable as *periodic* and *random*. The periodic effects (limit cycle oscillations) will be treated in Chapter 7. The random effects, *quantization noise*, are the subject of this chapter.

Several distinctions can be made when referring to quantization noise.

First, the storage registers (and quantizers) within a structure may have different, *nonuniform*, wordlengths; such a structure will always perform beter in terms of roundoff noise effects than the constrained case of uniform wordlengths [A11]. However, by using uniform wordlengths, the hardware expense and complexity will be greatly reduced. Often, little potential performance is lost by such a restriction. Since the A/D converter is usually a separate piece of hardware, little affected by the remaining compensator hardware architecture and design, it need not be subject to this restriction. Consequently, A/D and internal wordlengths can and typically do differ. We will assume that the signal variable registers are of uniform wordlength, and that the A/D wordlength can be different from the internal compensator wordlength.

The second distinction is in the placement of the structure's quantizers. On one hand, they can be inserted after *every* multiplication — ensuing adders would thus have to deal only with $n_r$-bit quantities. However, if we are willing to complicate the adders, quantization can be delayed until *after* the node additions, placing them just before each storage register or intermediate node value $r(k)$. With this method, adders would have to sum $n_r+n_c$-bit quantities, but fewer quantizers are needed. This alternative trades off hardware complexity (double- versus single-precision adders) for quantization noise (fewer quantizers implies fewer noise sources). Both these options will be considered in this chapter.

The final distinction in discussing quantization noise is in the *type* of quantizer used. Commonly, the choice is between *rounding*, which selects the finite-precision word that is closest to the ideal value, and *truncating*, which simply drops the extra bits of precision. Truncation, and specifically sign-magnitude truncation, has the advantage of requiring no extra hardware, and also an advantage

in terms of the resulting (reduced) number of possible limit cycle oscillations. However, rounding can be shown to have the advantage of reduced quantization noise effects, and the extra hardware it requires is not very complex. In addition, roundoff effects are more easily analyzed. Consequently, this chapter will primarily focus on roundoff quantization. In Chapter 7, we will consider other approaches to quantization which provide advantages in terms of limit cycle behavior, that is, fewer limit cycles or limit cycles of smaller amplitude.

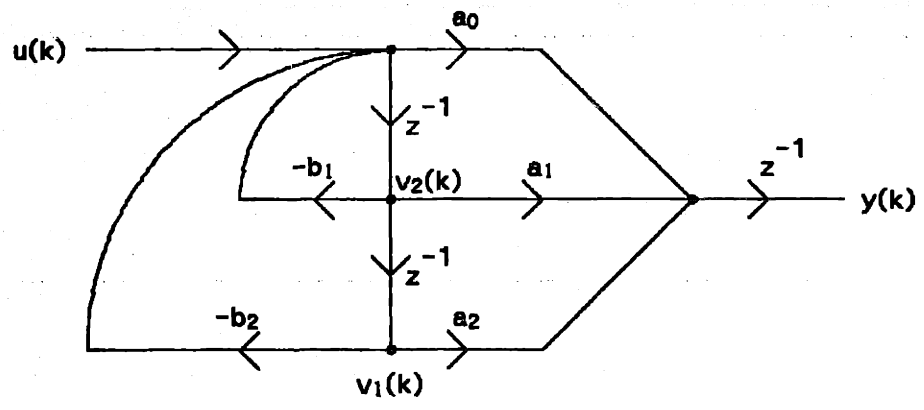
This chapter is organized as follows. Section 5.2 will discuss the major issue of dynamic range and scaling as applied to digital filters. In section 5.3 we will adapt these ideas for digital control compensator scaling. For this adaptation we will have to consider the entire closed-loop system in determining the appropriate scaling for compensators. Set-point LQG configurations and their implications as regards the scaling issue will also be discussed. Section 5.4 will describe the roundoff and sign-magnitude truncation quantization characteristics and present models for analyzing their effects. Methods of analyzing roundoff noise effects using the model developed in section 5.4 will be treated in section 5.5. Using these procedures, section 5.6 will describe the minimum roundoff noise filter structures introduced by Mullis and Roberts [18,37,38] and Hwang [39], and will then adapt these results to derive minimum roundoff noise compensator structures. Finally, section 5.7 will demonstrate the procedures developed in Chapter 5 for compensators by applying them to 10 candidate structures for implementing a specific control system.

## §5.2 Dynamic Range Constraints

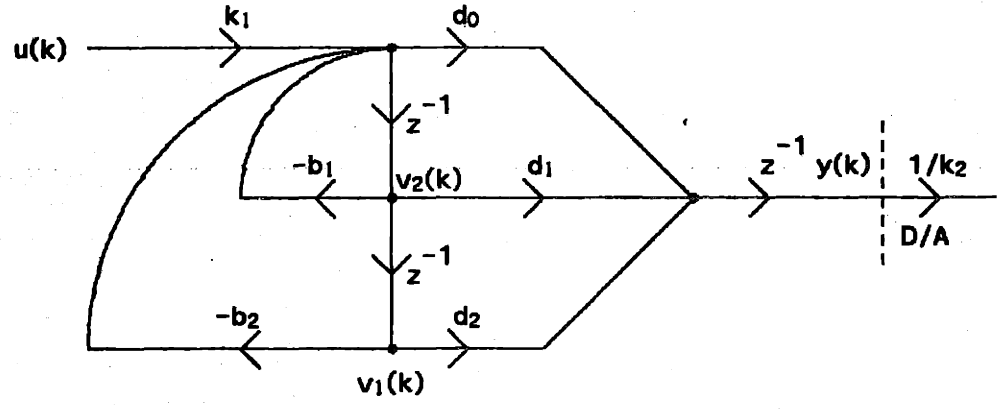
It is not meaningful to discuss quantization noise effects (proportional to the least significant bit of the signal word) without also considering the dynamic range of the signals within the structure. Our overall objective is to minimize the total number of bits necessary for the fixed-point digital words. Choosing a specific structure based on its required least significant bit size (quantization step size) is of little value unless the fixed-point words can represent the full dynamic range of the node signals while keeping overflows to a minimum. Thus, we must maximize signal-to-noise ratio without incurring overflow. These aims can be accomplished through *scaling*. By scaling the coefficients of a structure we can reduce the overall dynamic range of the signals within the structure and also normalize the maximum signal size (the overflow level) at each node. Once a structure is scaled, we *can* use the quantization step size as a valid basis for comparison with other structures which have been scaled using the same scaling procedure. (This section will present several of these scaling procedures.) Note that scaling does not alter the type of structure nor its ideal transfer function.

Consider the second-order filter of figure 5-1a. This structure has three states, implying three storage registers. Clearly, if the $v_2(k+1)$ node and the $y(k+1)$ output node do not overflow, then none of the node signals will overflow, since the other nodes are simply delayed versions of these two. Thus, scaling involves overflow constraints on these two nodes. Such constraints would be inequality constraints, that is, the signal magnitude must be *less* than the overflow level. Of course, too small a signal magnitude would result in higher quantization noise levels. Intuitively, we would like to alter the magnitudes of the signals at these two nodes just enough to prevent the occurrence of overflow, but without

(a) Unscaled




(b) Scaled




where $d_0 = a_0k_2/k_1$
$d_1 = a_1k_2/k_1$
$d_2 = a_2k_2/k_1$

Figure 5-1: Scaling a Second-Order Section

changing the filter transfer function. For example, to modify the signal magnitude at the $v_2(k+1)$ node, the input unity coefficient must be multiplied by some factor $k_1$, and then to preserve the transfer function of the filter, the three coefficients

$a_0$, $a_1$, and $a_2$ must be multiplied by $\frac{1}{k_1}$. Similarly, scaling the $y(k+1)$ node involves multiplying $a_0$, $a_1$, and $a_2$ by another factor $k_2$. The corresponding $\frac{1}{k_2}$ factor must then be absorbed by the output D/A converter to ensure an unchanged overall transfer function. The resulting scaled structure is shown in figure 5-1b.

An important choice must be made in selecting $k_1$ and $k_2$. Let us define *optimal* scaling to refer to that choice of scalers which satisfies the dynamic range constraints of the scaling procedure (inequality constraints) with equality. Thus, in general, such scalers will not be simple powers of two. For the example above, optimal scaling would result in a structure with 6 non-trivial multiplications, instead of 5. Optimal scaling usually carries the advantage over non-optimal scaling (which results when the scalers are constrained to be simple powers of two to simplify the hardware) of reduced quantization noise effects, even with the extra noise sources caused by the additional scaling coefficients. Thus, scaling introduces another tradeoff between performance and hardware complexity.

Two basic methods exist for choosing the dynamic range constraints. The first is a deterministic norm-based method introduced by Jackson [55]. Define the $L_p$ norm of a digital frequency-domain transform $H(z)$ as follows:

$$\|H\|_p = \left\{ \frac{1}{\omega_s} \int_0^{\omega_s} \left| H\left(e^{j\omega T}\right) \right|^p d\omega \right\}^{\frac{1}{p}} \qquad (5.1)$$

where $\omega_s$ is the sampling frequency in radians per second. If $F_i(z)$ is defined to be the transfer function from the input to the $i^{th}$ node that must be scaled, then

Jackson has used the fact that:

$$\left|r_l(k)\right| \le \left\|F_l\right\|_p \left\|U\right\|_{p_0} \quad \text{for } \frac{1}{p}+\frac{1}{p_0}=1\,, \quad p,p_0 \ge 1\,, \quad \text{and for all } k \qquad (5.2)$$

where $r_l(k)$ is the signal at the $l^{th}$ node to be scaled, and $U(z)$ is the z-transform of the filter input $u(k)$. Note that when this inequality is applied to $u(k)$ itself ($r_l(k)=u(k)$, $F_l(z)=1$) we find that $|u(k)| \le M_0$ if $\|u\|_{p_0} \le M_0$ for *any* $p_0 \ge 1$.

Now let us return to the scaling issue for node $l$. Assume that the maximum signal magnitude possible in the filter without overflow is $M_0$. Further assume that $\|U\|_{p_0} \le M_0$, and thus $u$ never overflows (its magnitude is always $\le M_0$). Then using (5.2), the node signal $r_l$ will not overflow if:

$$\left\|F_l\right\|_p \le 1 \quad \text{for all } l \qquad (5.3)$$

This scaling rule, $L_p$ scaling, must be satisfied at every node in the filter structure. Satisfying this rule with equality corresponds to *optimal* scaling as described above. For the example of figure 5-4, the scaling multipliers $k_1$ and $k_2$ must be chosen to satisfy (5.3) for $l = 1$ and $l = 2$.

The scaling rule described above still allows some degree of freedom even for optimal scaling, namely the choice of $p_0$ and $p$. If all we know about the input $u$ is that its magnitude will be below $M_0$ (so that $u$ could be a DC level), then $p_0$ can only be infinity. The only scaling that we can apply is $L_1$ scaling. However, assume that $u$ is also known to have no DC component, and in fact suppose that

$\|U\|_2 \le M_0$. Now we can select $p_0$ to have any value between 2 and infinity. For example, a $p_0$ of 2 would correspond to $L_2$ scaling, and a $p_0$ of infinity would correspond to $L_1$ scaling. In this case, we would select the scaling method that would result in lower levels of quantization noise, the $L_2$ scaling method. In general, the larger the $p$ (meaning smaller $p_0$), the less conservative the scaling rule will be, implying lower noise levels. Thus the more we know about the possible filter input signals, the better the scaling will be in terms of the resulting noise levels. For example, if all we know about the input is that it is smaller than $M_0$ in magnitude, then it could even be $u = M_0$. For this case, $p_0 = \infty$, and $p = 1$. Thus the $L_1$ norm of $F_i(z)$, the area under the $F_i(z)$ curve, must be forced to 1. This type of scaling is more conservative (results in more quantization noise) than $L_2$ or $L_\infty$ scaling.

A related deterministic scaling method has been described by Hwang [56]. This method is based on the time-domain $l_p$ norm of the infinite sequence $r_i(k)$ be defined as:

$$\|r_i\|_p = \left\{ \sum_{k=0}^{\infty} |r_i(k)|^p \right\}^{\frac{1}{p}} \tag{5.4}$$

The time-domain counterpart of (5.2) can be written as follows:

$$|r_i(k)| \le \|f_i\|_p \, \|u\|_{p_0} \quad \text{for } \frac{1}{p} + \frac{1}{p_0} = 1,\ p, p_0 \ge 1 \tag{5.5}$$

where $f_i(k)$ is the impulse response of node i at time $k$, and $u(k)$ is the filter input. The following scaling law results: if $M_0$ is the maximum signal magnitude al-

lowed in the filter and $\|u\|_{p_0} \le M_0$, then

$$\|f_l\|_p \le 1 \quad \text{for all } l \tag{5.6}$$

guarantees no overflow.

In order to compare $L_p$ and $l_p$ scaling methods, we must examine the relationship between the $L_p$ and $l_p$ norms [56]:

$$\|u\|_\infty \le \|u\|_1 \le \|u\|_2 = \|u\|_2 \le \|u\|_\infty \le \|u\|_1 \tag{5.7}$$

Given the relationship of (5.7), we can determine how conservative any given scaling rule is as compared to all the other scaling rules. From (5.7), we know that if the input satisfies the constraint $\|u\|_1 \le M_0$, then it must also satisfy $\|u\|_\infty \le M_0$ (but not vice-verse). Thus, as far as the type of input signal is concerned, knowing that the $L_1$ norm of the input is less than $M_0$ is *less* restrictive than knowing that its $l_\infty$ norm is less than $M_0$. We can generalize this statement to the entire list in (5.7). Since a less-restricted input corresponds to a more-conservative scaling, we can use the relationship (5.7) to determine how any scaling method compares to any other. Thus the most conservative scaling is $l_\infty$ scaling, and the least conservative corresponds to $l_1$ scaling. The actual scaling method selected will depend on what information is known about the filter input signal and its transform.

The second method for establishing dynamic range constraints and choosing scaling multipliers is a stochastic method [18,37,39]. With a random input signal, one considers the *probability of overflow* at each node rather than trying to

prevent overflow completely, which is no longer possible. Scaling will be accomplished by equalizing the probability of overflow at each node. Let us assume that the maximum signal level without overflow is $M_0$, and that the input is a zero-mean Gaussian random process of standard deviation $\frac{M_0}{3}$. The probability of overflow at the input A/D is then 0.003. The variance of the signal at node $i$ will be equal to $\frac{M_0}{3}\left(\sum_{k=0}^{\infty}\left[f_i(k)\right]^2\right)^{\frac{1}{2}}$. But this quantity is just the $l_2$ norm of $f_i(k)$ multiplied by the input variance. Thus, to *equalize* the probability of overflow at each node we must set $\|f_i\|_2 = 1$ for all $i$, which is equivalent to $l_2$ or $L_2$ deterministic (optimal) scaling.

In terms of a state-space structure as discussed in Mullis and Roberts [18,37], scaling corresponds to a *diagonal* similarity transformation of the unscaled structure. In the more general context of Chan's notation or the modified state space representation as described in Chapter 3, scaling can be described by a set of diagonal scaling matrices $S_i$. We will essentially follow the presentation of scaling for filters made by Chan [17], but in the context of the modified state space representation. (Thus a delay will be added to the output of the filter structure, as with a compensator, but the structure is still a filter — no external feedback is involved.) We will extend scaling ideas to the control setting in section 5.3.

A scaled structure has the following modified state space representation: (input $y$, output $u$)

$$\begin{bmatrix} \tilde{v}(k+1) \\ \tilde{u}(k+1) \end{bmatrix} = \tilde{\Psi}_q \cdots \Psi_1 \begin{bmatrix} \tilde{v}(k) \\ \tilde{u}(k) \\ y(k) \end{bmatrix} \tag{5.8}$$

where $v$ is a vector, $y$ and $\tilde{u}$ are scalars representing the compensator input and output respectively, and the tilde designates scaled quantities as opposed to the original unscaled values written without a tilde. The matrices $\tilde{\Psi}_q, \ldots, \tilde{\Psi}_1$ are related to the matrices $\Psi_q, \ldots, \Psi_1$ by:

$$\tilde{\Psi}_l = S_l \Psi_l \left(S_{l-1}\right)^{-1} \quad \text{for } l=q, \cdots, 1 \tag{5.9}$$

where

$$S_0 = \begin{bmatrix} S_q & 0 \\ 0 & 1 \end{bmatrix}$$

and all $S_l$ are diagonal. Since the $u(k)$ is scaled, the D/A scale factor must include an extra multiplicative factor $\rho$ equal to the reciprocal of the $(n+1,n+1)^{th}$ entry of $S_q$ to convert $\tilde{u}(k)$ to $u(k)$.

In the context of the modified state space representation, we can now examine stochastic $l_2$ scaling using (5.9). Let us partition $\Psi_\infty = \Psi_q \cdots \Psi_1$ (defined in section 3.3) as follows:

$$\Psi_\infty = \begin{bmatrix} \Psi_{11} & \Psi_{12} \end{bmatrix} \tag{5.10}$$

where $\Psi_{11}$ is $(n+1)\times(n+1)$ and $\Psi_{12}$ is $(n+1)\times 1$. Assuming infinite-precision coefficients, the states, input, and output of the filter can be related with the following state space of order $n+1$:

$$\begin{bmatrix} v(k+1) \\ u(k+1) \end{bmatrix} = \Psi_{11} \begin{bmatrix} v(k) \\ u(k) \end{bmatrix} + \Psi_{12} y(k)$$
$$u(k) = [0 \; 0 \; 0 \; \cdots \; 0 \; 1] \begin{bmatrix} v(k) \\ u(k) \end{bmatrix} \tag{5.11}$$

For this system of equations, the state covariance matrix $V$ can be written:

$$V = \overline{\begin{bmatrix} v(k) \\ u(k) \end{bmatrix} \begin{bmatrix} v'(k) \; u'(k) \end{bmatrix}} = \epsilon \sum_{j=0}^{\infty} \left( \Psi_{11}^{j-1} \Psi_{12} \right) \left( \Psi_{11}^{j-1} \Psi_{12} \right)' \tag{5.12}$$

Let us define the matrix $K_q$ to be $V/\epsilon$. A Lyapunov equation equivalent to (5.12) is usually easier to evaluate for computing $K_q$:

$$\Psi_{11} K_q \Psi_{11}' + \Psi_{12} \Psi_{12}' = K_q \tag{5.13}$$

The diagonal elements of $K_q$ represent the gains from the input variance to the state node variances. Now we need the gains from the input node to the intermediate node variances, assuming that the structure is multi-level. Since the intermediate nodes are related to the state nodes via the precedence level matrices $\Psi_1$ through $\Psi_{q-1}$, we can compute a set of matrices $K_i$ whose diagonal elements are the desired gains from the input variance to the variances of the intermediate node vector $r_i$:

$$K_i = \Psi_i \Psi_{i-1} \cdots \Psi_1 \begin{bmatrix} K_q & 0 \\ 0 & 1 \end{bmatrix} \Psi_1' \cdots \Psi_{i-1}' \Psi_i' \quad \text{for } i=1, \cdots, q-1 \tag{5.14}$$

Stochastic scaling ($l_2$ scaling), which equalizes the probability of overflow at all the nodes in the structure including the input, can be realized by forcing all the diagonal entries of the $K_i$ matrices to unity. Thus all the node variances will be

the same as the input variance. This scaling is accomplished by applying a diagonal transformation to the unscaled structure where:

$$\left[S_i\right]_{jj} = \left(\left[K_i\right]_{jj}\right)^{-\frac{1}{2}} \quad \text{for } i=1, \cdots, q \text{ and all } j \tag{5.15}$$

The resulting structure (5.8) would have $\tilde{K}_i$ matrices whose diagonal elements were all unity entries, as desired.

## §5.3 Digital Feedback Compensator Scaling

In this section we will discuss the implications of LQG set-point configurations to the issue of compensator scaling, and then adapt the $l_2$ stochastic scaling method described in the previous section for filters to the digital feedback compensator.

The scaling issue for digital compensators differs in certain respects from the filtering applications described above. The first of these involves the type of scaling appropriate to LQG systems. Most of the LQG configurations as described in Chapter 2 will have *set points*, in other words, reference inputs for the regulator portion of the design. These non-zero set-point regulators [1] will have the same parameter values as described in Chapter 2, independent of the set point, but the resulting DC compensator input will affect the scaling. As stated before, conservative scaling is required whenever we allow the presence of DC inputs. Specifically, $l_2$ scaling is not possible, eliminating the stochastic approach.

Figure 5-2 presents the set-point LQG system described in Kwakernaak and Sivan [1], where $u_r$ is the reference input. If we wish to drive the output $y$ to $y_r$, then $u_r$ must be set to $H_c^{-1}(1)\,y_r$, where $H_c(z)$ is the closed-loop transfer

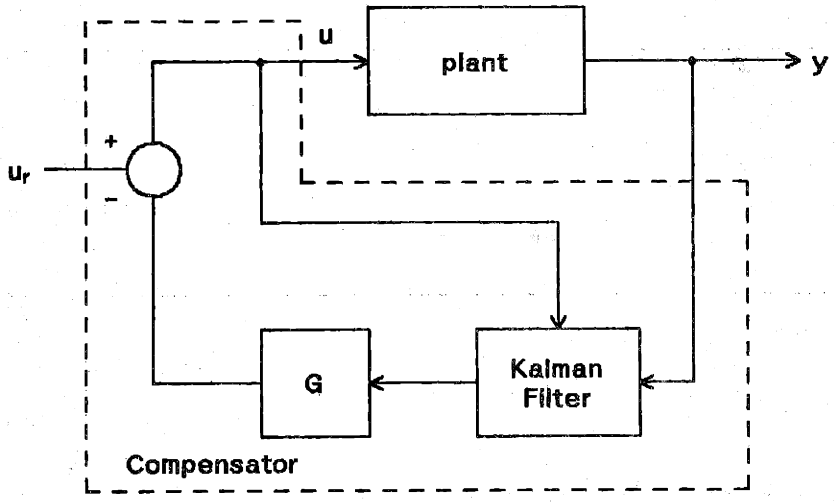



Figure 5-2: Set-Point Compensator Configuration

function from $u_r$ to y:

$$H_c(z) = L(zI - \Phi + \Gamma G)^{-1}\Gamma \tag{5.16}$$

Unfortunately, this compensator has a DC input since the steady-state value of y is non-zero. Thus $l_2$ scaling is not possible. However there is one other (equivalent) approach to describing the system of figure 5-2 and the equations of Chapter 2. Define $\xi$, $\nu$, and $\gamma$ to be the *deviations* of the states, input, and output from the steady-state values $x_0$, $u_0$, and $y_0$. Thus, $\xi = x - x_0$, $\nu = u - u_0$, and $\gamma = y - y_0$. As in [1], the following relationship must hold:

$$\begin{aligned} x_0 &= \Phi x_0 + \Gamma u_0 \\ y_0 &= L x_0 \end{aligned} \tag{5.17}$$

Now, follow through the LQG design equations of Chapter 2 for the (deviations of

the) states $\xi$, input $\nu$, and output $\gamma$. With the *actual* state, input, and output variables being represented by $x$, $u$, and $y$, we can then produce figure (5-3). Thus it is possible to use an alternate LQG set-point configuration where the

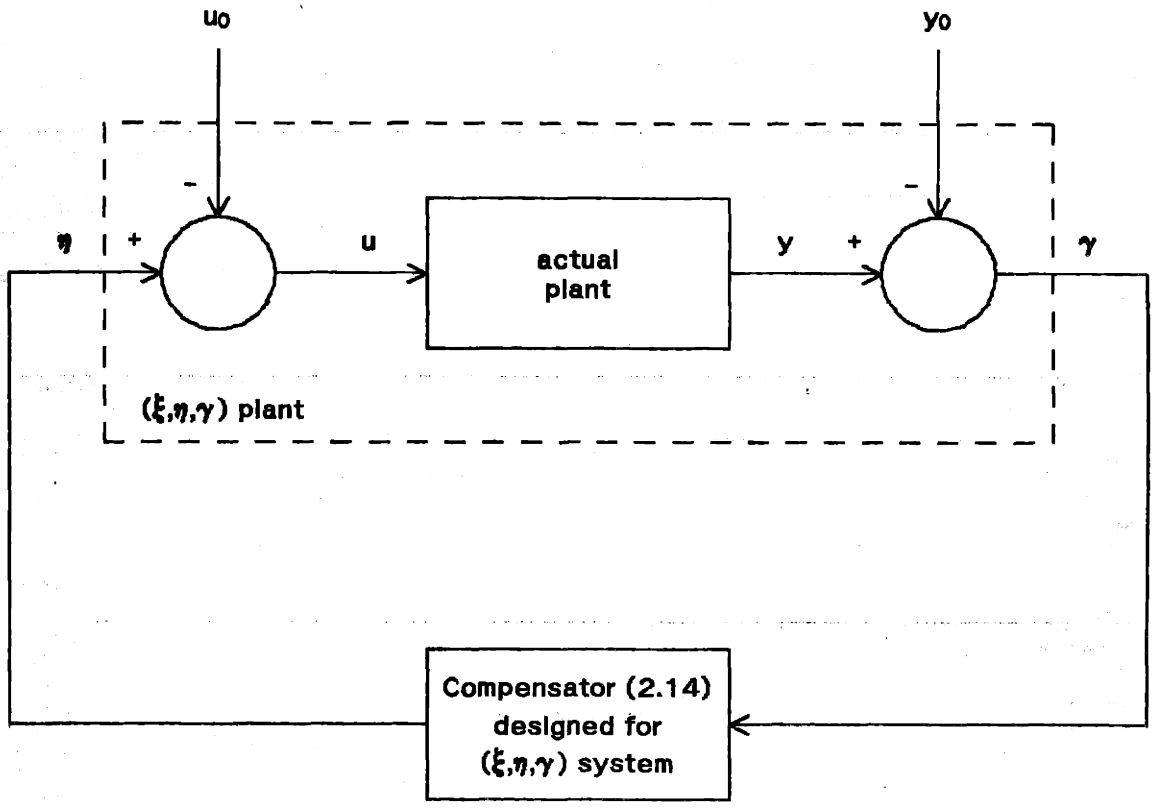



Figure 5-3: Alternate LQG Set-Point Configuration

compensator input has an average value of zero, thereby allowing us to apply stochastic ($l_2$) scaling. The disadvantage to this alternate configuration is the necessity of having *two* reference inputs which must maintain the precise relationship (5.17), typically in the presence of plant parameter uncertainty.

This disadvantage will vanish whenever the plant has a series integration (at least one pole at the origin s=0), which is a very common occurrence in con-

trol systems. In fact, frequently an integrator is *added* to an actuator (part of the plant) to provide desensitivity to constant disturbances. To see the effect of an integrator pole on the configuration of figure 5-3, let us write $u_0$ as $\left(L(I-\Phi)^{-1}\Gamma\right)^{-1} y_0$. However, since the DC gain $L(I-\Phi)^{-1}\Gamma$ blows up if there are any open-loop integrator poles in the plant (poles at $z=1$), $u_0$ is forced to zero. In other words, if the plant has any series integration, the LQG configuration of figure 5-3 need have only one reference input, $y_0=y_r$, and not two. Note that the configuration of figure 5-2 does not change when the plant has integrator poles; both compensator inputs will still have DC components, and the system as a whole still requires the reference input $u_r$. From this point on, the figure 5-3 configuration is assumed so that $l_2$ scaling can be applied.

The second difference between filter and compensator scaling arises when we try to apply $l_2$ scaling as described in (5.12)-(5.15) to a compensator. This procedure would treat the compensator as a separate entity (functionally a filter), ignoring the LQG plant and feedback path. Yet a compensator operating open-loop need not even be stable. The stochastic scaling method requires the variances of the signal variables at the compensator state nodes so that the matrices $K_l$ and $S_l$ can be computed. Clearly these variances depend on the overall closed-loop performance. Thus we will have to *adapt* the filter scaling procedure so that it applies to digital feedback compensator scaling.

We have developed the following scaling procedure to account for the LQG feedback system in which the compensator is embedded. The steady-state variances of the $n$ plant states and $n+1$ compensator states can be found by combin-

ing the state and compensator equations into a single augmented state space:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \\ u(k+1) \end{bmatrix} = A \begin{bmatrix} x(k) \\ v(k) \\ u(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ \Psi_{12} w_2(k) \end{bmatrix} \tag{5.18}$$

where

$$A = \left[\begin{array}{c|c|c} \Phi & 0_n & \Gamma \\ \hline \Psi_{12} L & \Psi_{11} & \end{array}\right]$$

and $0_n$ represents an all-zero $n \times n$ matrix and $\Psi_{11}$, $\Psi_{12}$ represent the unscaled compensator as partitioned in (5.10). With this state space, let us now follow the general scaling procedure outlined in section 5.2. The overall $(2n+1)\times(2n+1)$ state covariance matrix $Z$ can be computed by solving the following discrete-time Lyapunov equation: [16]

$$Z = A\, Z\, A' + C \tag{5.19}$$

where

$$C = \begin{bmatrix} \Theta_1 & 0 \\ 0 & \left\{ \Psi_{12} \Theta_2 \Psi'_{12} \right\} \end{bmatrix}$$

We now partition $Z$ to separate the plant and compensator covariances:

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12}' & Z_{22} \end{bmatrix} \tag{5.20}$$

where $Z_{11}$ is $n \times n$. As defined in (5.12), $K_q$ will result from dividing $Z_{22}$ by

$\overline{yy} = \sigma_y^2$. Thus

$$K_q = \frac{Z_{22}}{LZ_{11}L'} \tag{5.21}$$

To compute $K_l$ as in (5.14), the compensator states and input $y$ must be uncorrelated. However, feedback introduces correlation:

$$E\left\{\begin{bmatrix} v \\ u \\ y \end{bmatrix}\begin{bmatrix} v' & u' & y' \end{bmatrix}\right\} = \begin{bmatrix} Z_{22} & Z_{12}'L' \\ LZ_{12} & LZ_{11}L' \end{bmatrix} \tag{5.22}$$

Normalizing by $\sigma_y^2$, we get:

$$K_l = \Psi_l\Psi_{l-1}\cdots\Psi_1\begin{bmatrix} K_q & \dfrac{Z_{12}'L'}{LZ_{11}L'} \\ \dfrac{LZ_{12}}{LZ_{11}L'} & 1 \end{bmatrix}\Psi_1'\cdots\Psi_{i-1}'\Psi_i' \tag{5.23}$$

$$\text{for } l=1,\cdots,q-1$$

The scaling matrices $S_l$ now follow directly from (5.15). This scaling technique has been applied for the optimal $l_2$ scaling of the compensator structures treated in this thesis.

The last controller scaling question that arises concerns the A/D and DA converter scale factors. Once a compensator is scaled via (5.18)-(5.23), the probability of overflow within the compensator equals the probability of overflow at the

A/D (for Gaussian A/D inputs). By setting the A/D scale factor (and inversely adjusting the D/A scale factor), we can control this overflow probability. In such a procedure the *compensator* scaling procedure is unaffected by the A/D scale factor — the scaling multipliers remain invariant. The dynamic range of the input and output transients in the system (caused by changing the set point for example) and of the set point itself will also affect the actual A/D scaling choice. Whatever is chosen for $k_{ad}$ (and $k_{da}$ must include a $k_{ad}^{-1}$ factor as well as the $\rho$ factor resulting from the scaling of the compensator output node) the effect of quantization noise on the performance index or on the output noise variance will increase as $k_{ad}^{-2}$.

## §6.4 Quantizer Characteristics and Models

In order to analyze the effects of quantization in some tractable and systematic fashion it is necessary to model the nonlinear operation of quantization. This section will present the roundoff and sign-magnitude truncation quantizer input-output characteristics and the models commonly used for them. A discussion of model validity then follows. We will assume throughout that the fixed-point words representing signal variables have $n_f$ bits to the right of the binary point, and that $\Delta$ is defined to be the quantization step size. ($\Delta = 2^{-n_f}$)

Figure 5-4 shows the input-output characteristic of the roundoff quantizer. Let $RO(x)$ be the rounded value of $x$. The error associated with such a quantizer, $e = x - RO(x)$, satisfies (5.24):

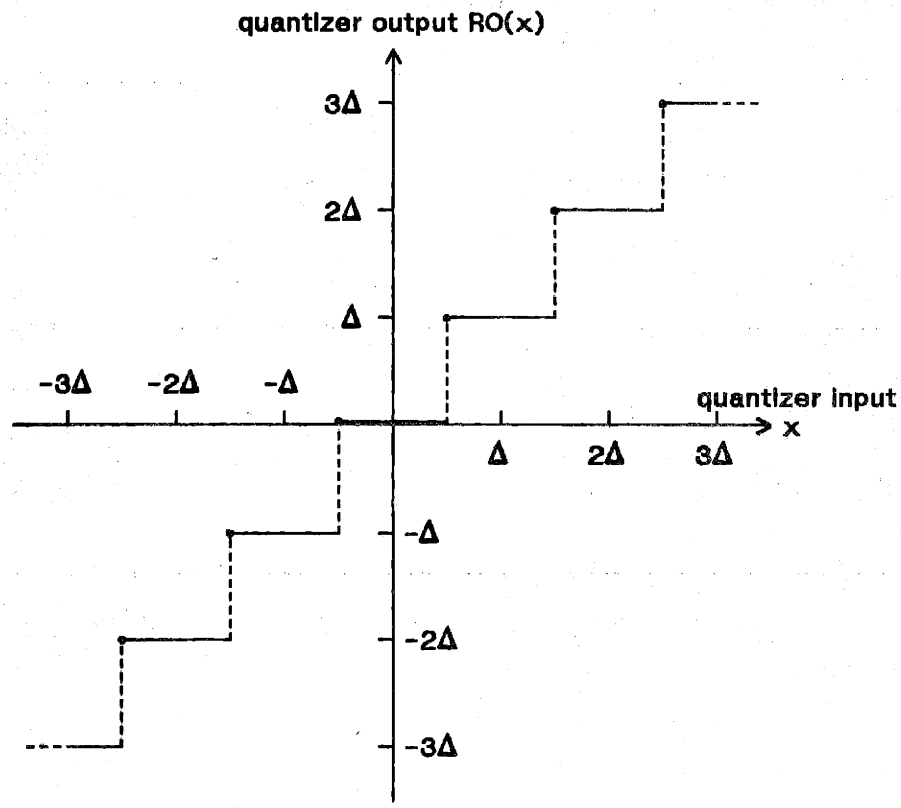



Figure 5-4: Nonlinear Roundoff Characteristic

$$-\frac{\Delta}{2} < e \leq \frac{\Delta}{2} \tag{6.24}$$

The model commonly used to represent the roundoff quantization operation is the additive white noise model [57]. In this case, roundoff is modelled *linearly* as a zero-mean random noise added to the ideal (infinite-precision) signal value. The noise e is assumed to have a uniform density as shown in figure 5-5 and to be uncorrelated with the quantizer input signal. The validity of this model is an important consideration, since its use simplifies quantization noise analysis a great deal. For a continuous-time quantizer input signal, the usually-applied rule of thumb states that the noise model is valid if the input to the quantizer crosses

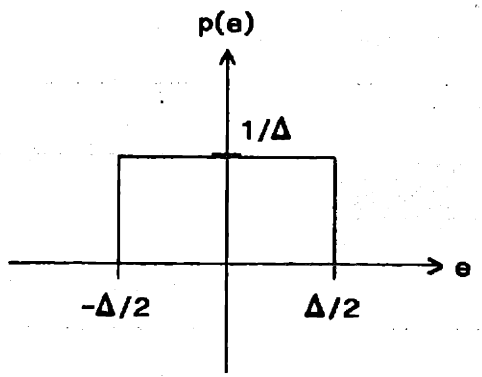



Figure 5-5: White Noise Error Model Density

'many' quantization levels between sample times [28] — that is, the input magnitude must fluctuate over a range >> Δ in each $T$ second period.

A detailed analysis of the validity of the additive noise roundoff model has been carried out by Sripad and Snyder [58] and Sripad [13]. These authors have established necessary and sufficient conditions on the quantizer input such that the model is *exact*. Let $\phi_x(s)$ be the characteristic function of the quantizer input $x$ (the Laplace transform of the probability density $p(x)$). Then:

(1) The density $p(e)$ matches that of figure 5-5 if and only if $\phi_x\left(\frac{2\pi l}{\Delta}\right) = 0$ for $l \neq 0$ and $l$ an integer.

(2) The noise samples $e(k)$ and $e(k+1)$ are uncorrelated if and only if the joint characteristic function between the two inputs $x(k)$ and $x(k+1)$ satisfies $\phi_{x(k),x(k+1)}\left(\frac{2\pi j}{\Delta},\frac{2\pi l}{\Delta}\right) = 0$ for all $j$, $l \neq 0$.

(3) The quantities $e(k)$ and $x(k)$ are uncorrelated if and only if $\phi_x\left(\frac{2\pi l}{\Delta}\right) = 0$

and $\frac{d}{d\omega}\phi_x(\omega) = 0$ for $\omega = \frac{2\pi l}{\Delta}$, and all $l \neq 0$.

Unfortunately these conditions are difficult to verify since the probability density function of *every* quantizer input must be known! Even so, if a quantizer unput contains any Gaussian noise (typically assumed in control problems, at least for the A/D input) then *none* of the above conditions hold exactly.

This validity restriction is not as serious as it seems. Sripad [13] has investigated the properties of the quantization error given a Gaussian input of variance $\sigma^2$. From these results it is evident that the error $e(k)$ has an approximately uniform distribution for $\sigma \gtrsim .7\Delta$, a condition that is not particularly restrictive.

In considering multiple quantizers (which is the usual case), the question of the interaction of the quantization errors arises. The above analysis actually applies to a single quantizer only. When the model is used for all the quantizers within a complex (recursive) structure, we further assume that all such noise sources are independent. The question of the validity of this assumption is even more complex. However, it *can* be said that as a general technique, the additive noise model has proven itself quite useful for the analysis of roundoff noise effects in digital filters. Furthermore, any analysis techniques aimed at selecting wordlengths based on the effects of quantization noise need not be exact anyway — the internal and A/D wordlengths can only be selected in units of whole bits. When the roundoff noise model breaks down, it tends to do so in a major way; limit cycles occur. These oscillations are usually quite evident when they are present (see Chapter 7). For our analyses, however, we *will* assume that the uncorrelated additive white noise model applies.

Sign-magnitude truncation refers to the quantization operation of simply

dropping the extra bits of precision in the quantizer input. The advantages to this type of quantization are its simplicity — no extra hardware is required to implement sign-magnitude truncation, unlike the roundoff case, and this type of quantization gives rise to fewer limit cycles. Figure 5-6 shows the input-output

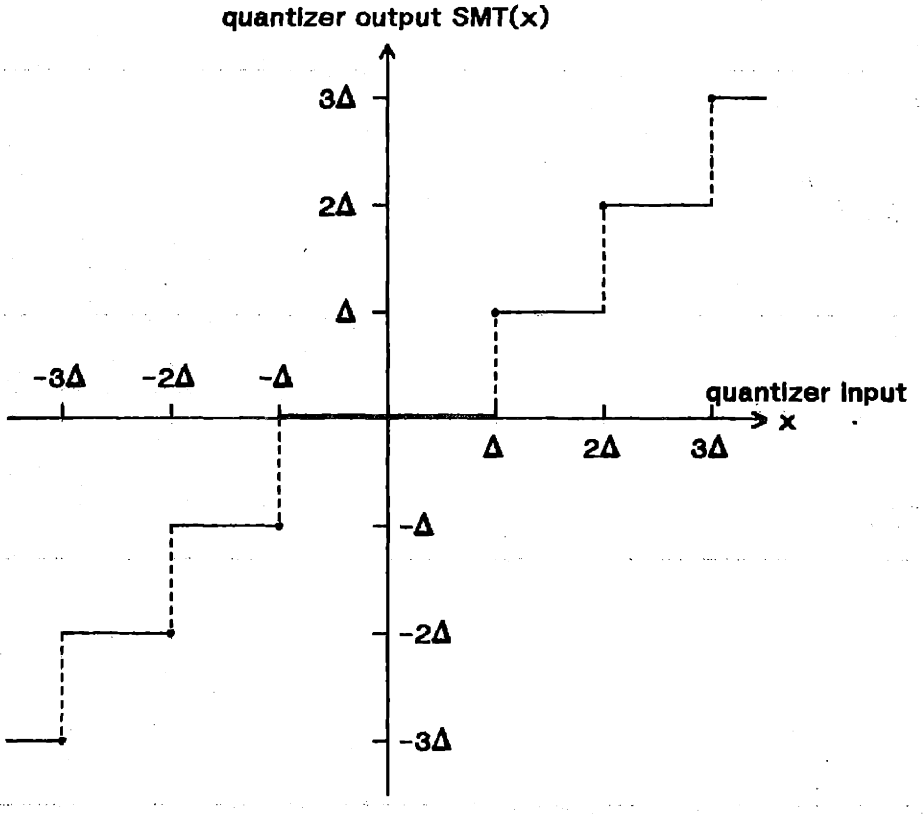



Figure 5-6: Nonlinear Sign-Magnitude Truncation Characteristic

characteristic of this quantizer. The quantization errors are now bounded as follows;

$$\begin{aligned} 0 \le e < \Delta \quad & \text{for } x \ge 0 \\ -\Delta < e \le 0 \quad & \text{for } x \le 0 \end{aligned} \tag{5.25}$$

For this type of quantization, the modelling problem is more difficult. From (5.25)

we can see that a definite correlation exists between the error and the input values with sign-magnitude truncation; $e(k)$ is a function of the sign of the quantizer input $x(k)$. Such noise in a digital structure is termed *state-dependent* noise. Although Sripad [13] does present an additive white model for this quantization operation, the conditions for which the model is valid are too restrictive for general application. The additive white noise model is not even approximately valid, as is the roundoff model. Claasen, Mecklenbräuker, and Peek [59] have proposed a quasi-linear model for sign-magnitude truncation:

$$SMT(x) \approx x - \frac{\Delta}{\sqrt{2\pi}\sigma} x + e \tag{5.26}$$

where $e$ is an uncorrelated white noise of variance $\left(\frac{1}{3} - \frac{1}{2\pi}\right)\Delta^2$ and the quantizer input $x$ is assumed to be a Gaussian process. The dependence on $\sigma$ (the variance of $x$) accounts for the quasi-linearity and also the complexity in using this model for analysis, since the variance of each quantizer input must be computed. An efficient technique for evaluating these variances is given in [59].

Empirically, the noise variance at the output of a digital filter using sign-magnitude truncation would typically be about 5 to 10 times that of the same filter using roundoff quantization [60]. Thus one should have an extra two bits per signal word when using sign-magnitude truncation in order to produce the same (or better) noise performance as would result from using roundoff quantization. Beyond this qualitative statement, we will not consider the specific analysis of sign-magnitude quantization noise effects for control compensators.

## §5.5 Roundoff Noise Analysis

This section will examine several methods for evaluating the effects of quantization noise in digital filters and compensators. As mentioned before, we will focus on *roundoff* quantization. For filtering applications, we are typically concerned with the statistical effects of quantization on the filter output. Although Jackson [61] examines various norms of the output noise spectrum, the noise variance (the $L_2$ norm squared) is usually taken to be the metric.

There are two basic methods for computing the output variance resulting from quantization noise effects, one in the frequency domain and one in the time domain. The frequency-domain analysis method is an application of residue theory [62]. Given the $i^{th}$ noise source of variance $\frac{\Delta^2}{12}$ and the (scaled structure) transfer function $G_i(z)$ from the noise source to the output node, then the output variance $\sigma_i^2$ due to this noise source can be written:

$$\sigma_i^2 = \frac{\Delta^2}{12(2\pi j)} \oint G_i(z)G_i(z^{-1})z^{-1}dZ \qquad (5.27)$$

where $j$ represents the square root of -1. The contour integral (5.27) can be evaluated by factoring $G_i(z)G_i(z^{-1})z^{-1}$ to determine its pole locations. If $n_p$ of these poles $z_j$ lie *inside* the unit circle, then

$$\sigma_i^2 = \frac{\Delta^2}{12} \sum_{j=1}^{n_p} \left( \text{Residue} \left\{ G_i(z)G_i(z^{-1})z^{-1} \right\} \text{at } z_j \right) \qquad (5.28)$$

Since every noise source is assumed to be uncorrelated with every other, the total output variance will simply be the sum of all the $\sigma_i^2$.

If we apply this residue method to the A/D quantization noise source, we see that $\sigma_{ad}^2$ will depend on the filter transfer function $H(z)$. Since $H(z)$ is independent of the structure chosen, given infinite precision coefficients, the effect of the A/D roundoff noise on filter output variance is dependent only on $k_{ad}$ and the A/D wordlength. For a compensator the effect of A/D roundoff noise on $J$ is also structure-independent, given infinite-precision coefficients.

The time-domain approach to analyzing roundoff effects is presented by Hwang [63] for one-level state space structures and Chan [17] for the general multi-level case. In the context of the modified state space representation as presented in (3.19), the derivation proceeds as follows. Assume that the structure has already been scaled so that the factor $\rho$ described after equation (5.9) must be included to produce $u(k)$ from the scaled $\tilde{u}(k)$. For a filter of input $y$, scaled output $\tilde{u}$, and scaled states $\tilde{v}$, the effect of roundoff noise on the filter states can be described by:

$$\begin{bmatrix}\tilde{v}(k+1)\\ \tilde{u}(k+1)\end{bmatrix} = \Psi_{11}\begin{bmatrix}\tilde{v}(k)\\ \tilde{u}(k)\end{bmatrix} + \sum_{l=2}^{q}\Psi_q \cdots \Psi_l \epsilon_{l-1}(k) + \epsilon_q(k) + \Psi_{12}\epsilon_{ad}(k) \quad (5.29)$$

where $\epsilon_l(k)$ represents the noise sources due to the product quantizations associated with the precedence level matrix $\Psi_l$, and $\epsilon_{ad}(k)$ represents the A/D noise source. Recall that all such error sources are assumed to be uncorrelated. Thus, the roundoff noise covariances can be written:

$$\overline{\epsilon_l(k)\,\epsilon^t{}_l(k)} = \frac{\Delta_r^2}{12}\Lambda_l$$

$$\overline{\epsilon_{ad}^2} = \frac{\Delta_{ad}^2}{12} \qquad (5.30)$$

where $\Delta_r$ is the internal quantization step size of the structure, $\Delta_{ad}$ is the A/D quantization step size, and $\Lambda_l$ is a diagonal matrix whose $(j,j)^{th}$ entry equals the number of non-integer coefficients in the $j^{th}$ row of $\Psi_l$, that is, the number of roundoff error sources associated with the $j^{th}$ component of $r_l$. This expression assumes that roundoff occurs after every non-trivial product. If double-precision adders are used as described in section 5.1, then simply replace all the non-zero entries of $\Lambda_l$ in (5.30) with ones.

To use (5.29) and (5.30) in computing the output variance, we can take either of the approaches used in section 5.2 for computing variances; that is, either the infinite series of (5.12) or the Lyapunov equation of (5.13) can be used. For the infinite-series approach, we would have to approximate the series by computing only a finite number of terms. The closer to the unit circle any of the poles of the system (5.30) are, the more terms will be required for an acceptable approximation [63]. Consequently, we will use the Lyapunov equation method.

The steady-state (scaled) state covariance matrix $\tilde{V}$ can be computed by solving the following Lyapunov equation:

$$\tilde{V} = \Psi_{11}\,\tilde{V}\,\Psi_{11}{}' + \Omega \qquad (5.31)$$

where

$$\Omega = \tilde{\Psi}_{12} \frac{\Delta_{ad}^2}{12} \tilde{\Psi}'_{12} +$$

$$\frac{\Delta_r^2}{12}\left\{\Lambda_q + \tilde{\Psi}_q \Lambda_{q-1} \tilde{\Psi}'_q + \tilde{\Psi}_q \tilde{\Psi}_{q-1} \Lambda_{q-2} \tilde{\Psi}'_{q-1} \tilde{\Psi}'_q + \ldots + \tilde{\Psi}_q \ldots \tilde{\Psi}_2 \Lambda_1 \tilde{\Psi}'_2 \ldots \tilde{\Psi}'_q\right\}$$

The output variance of $\tilde{u}$ will simply be equal to the lower right-hand corner entry of $\tilde{V}$. Note that the above equations for roundoff analysis are solved using infinite-precision coefficients for simplicity. The insertion of the actual finite-wordlength coefficients would only change the results in a minor way. (In this case, there will be also a slight dependence on structure for the A/D noise contribution.) The use of infinite-precision coefficients is especially justified when one recalls that the selection of an internal or A/D wordlength can only be made in terms of whole bits.

Now let us adapt this approach for the digital feedback compensator. Again, we need to consider the behavior of the closed-loop system, as done by Knowles and Edwards [7] and Curry [8] for sampled-data systems and Sripad [13]. Curry [8] has considered the second moment of the system output error due to rounding for a specific sampled-data control system with a direct form II compensator structure. Knowles and Edwards [7] also used the additive white noise model for generating a bound on the quantization noise effects of direct form II, cascade, and parallel compensator structures. Sripad [13] considered the increase in the performance index $J$ due to roundoff, using the additive white noise model, but did not consider either the scaling issue or an accurate and general notion of a compensator structure. Our results will be more general since we can consider any type of compensator structure, and they will of course be

adapted from the digital filtering approach described above. The factors $\rho$ and $k_{ad}$ described in section 5.2 must now be now explicitly included in the analysis procedure. The scaled, augmented plant/compensator system, including roundoff noise sources (but not plant or measurement noises), can be written:

$$\begin{bmatrix} \tilde{x}(k+1) \\ \tilde{v}(k+1) \\ \tilde{u}(k+1) \end{bmatrix} = \tilde{A} \begin{bmatrix} \tilde{x}(k) \\ \tilde{v}(k) \\ \tilde{u}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ e_q(k) + \sum_{l=2}^{q} \Psi_q \cdots \Psi_l e_{l-1}(k) + \Psi_{12} e_{ad}(k) \end{bmatrix} \quad (5.32)$$

where

$$\tilde{A} = \left[\begin{array}{c|c|c} \Phi & 0_n & \Gamma k_{da} \\ \hline \Psi_{12} L k_{ad} & \Psi_{11} & \end{array}\right] \quad \text{and } k_{da} = \frac{\rho}{k_{ad}}$$

The resulting (scaled) state covariance matrix $\tilde{Z}$ (due only to roundoff noise) will be the solution to the following Lyapunov equation:

$$\tilde{Z} = \tilde{A}\, \tilde{Z}\, \tilde{A}' + \begin{bmatrix} 0 & 0 \\ 0 & \Omega \end{bmatrix} \quad (5.33)$$

The covariance matrix $\tilde{Z}$ can be related to the performance index $J$ by using the *trace* form of $J$ , equivalent to (2.6):

$$J = \text{trace}(Q\overline{xx'}) + 2\,\text{trace}(M\,\overline{ux'}) + \text{trace}(R\,\overline{uu'}) \quad (5.34)$$

$$= \text{trace}\, \Upsilon Z$$

where

$$\Upsilon = \begin{bmatrix} Q & 0_n & M \\ 0_n & 0_n & 0 \\ M' & 0 & R \end{bmatrix}$$

By solving (5.33) and evaluating (5.34) for the scaled system covariance matrix $\tilde{Z}$ we can compute the increase $dJ$ due to *roundoff noise alone*. Again, the infinite-precision coefficient values of the structure are used.

The analysis procedure described above extends easily to multiple-input multiple-output structures, but as described in Chapter 9, the scaling issue is more complex.

## §5.6 Minimum Roundoff Noise Structures

Now that an analytic technique for treating roundoff noise effects has been presented, both for digital filters and for digital compensators, we can describe minimum roundoff noise structures. (See Chapter 3.) First, we will present the one-level minimum roundoff noise *filter* structure derived by Mullis and Roberts [18,37,38] and Hwang [39], and then we will adapt the technique to produce a one-level minimum roundoff noise compensator structure. Assume that a one-level *filter* structure has been $l_2$ scaled using (5.8)-(5.15), and that the roundoff noise could be evaluated with (5.31). (Neglect A/D noise.) For one level, (5.31) can be rewritten to include scaling:

$$\tilde{V} = S_1 \Psi_{11} S_1^{-1} \tilde{V} (S_1)^{-1} \Psi_{11}' S_1 + \frac{\Delta_r^2}{12} \Lambda_1 \qquad (5.35)$$

Recall that $\Lambda_1$ is a diagonal matrix whose $j^{th}$ diagonal entry equals the number of roundoff error sources represented in the $j^{th}$ row of $\Psi_1$, and that the scaling ma-

trix $S_1$ is diagonal. The output variance due to product quantization can be expressed with the following trace:

$$\sigma_0^2 = \rho^2 \,\text{trace}\, \Pi\, \tilde{V} \tag{5.36}$$

where

$$\Pi = \begin{bmatrix} 0 & \cdots & 0 \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ 0 & \cdots & 1 \end{bmatrix}$$

By substituting $V$ for $S_1^{-1}\tilde{V}(S_1)^{-1}$, we can rewrite (5.35) and (5.36):

$$V = \Psi_{11} V \Psi_{11}' + \frac{\Delta_r^2}{12}(S_1)^{-1}\Lambda_1(S_1)^{-1}$$

$$= \Psi_{11} V \Psi_{11}' + \frac{\Delta_r^2}{12}\Lambda_1 S_1^{-2} \tag{5.37}$$

$$\sigma_0^2 = \rho^2 \,\text{trace}\, (\Pi\, S_1\, V\, S_1)$$

$$= \text{trace}\, \rho^2 S_1\, \Pi\, S_1\, V$$

$$= \text{trace}\, (\Pi\, V) \tag{5.38}$$

Using the theory of adjoint operators [1], computing $\sigma_0^2$ via (5.37) and (5.38) is exactly equivalent to solving the following adjoint Lyapunov equation and evaluat-

ing the trace of (5.40): (See Appendix B)

$$W_1 = \Psi_{11}' W_1 \Psi_{11} + \Pi \tag{5.39}$$

$$\sigma_0^2 = \frac{\Delta_r^2}{12} \, trace \, (\Lambda_i \, S_i^{-2} \, W_1)$$

$$= \frac{\Delta_r^2}{12} \sum_{i=1}^{n+1} [\Lambda_i]_{ii} [K_1]_{ii} [W_1]_{ii} \tag{5.40}$$

This alternate expression for roundoff noise will be important in the development of an iterative constrained optimization technique for minimizing roundoff noise effects, both for filter structures (see Chan [17]) and for compensator structures (see Chapter 8).

Using the expression in (5.39), and the Lyapunov equation (5.13) for $K_1$, Mullis and Roberts [18] and Hwang [39] present a method for determining the structure that minimizes $\sigma_0^2$. The matrix $\Lambda_1$ is assumed to be the identity $I$ (for double-precision adders) or $(n+1)I$ (for the case of single-precision adders and $n+1$ coefficients). Since the $i=(n+1)^{th}$-term in the summation expression for $\sigma_0^2$ in (5.40) is not alterable by a similarity transform, we can ignore it for now and deal only with $K$ and $W$, the upper $n \times n$ portions of $K_i$ and $W_i$. Thus we must minimize the following sum:

$$\sum_{i=1}^{n} K_{ii} W_{ii} \tag{5.41}$$

If $P$ is an $n \times n$ (similarity) transformation matrix, then the product $KW$ can be

shown to transform to $P^{-1}KWP$. Thus the $n$ eigenvalues of $(P^{-1}KWP)$ are invariant under transformation by $P$. These eigenvalues are called the second-order filter modes $\mu_i^2$. Mullis and Roberts [37] prove the following inequality. If $K$ and $W$ are $n \times n$, symmetric positive-definite matrices, then

$$\frac{1}{n}\sum_{i=1}^{n} K_{ii}W_{ii} \geq \left[\frac{1}{n}\sum_{i=1}^{n} \mu_i\right]^2 \tag{5.42}$$

An (optimal) transformation exists such that the transformed $K_t$ and $W_t$ ($K_t = P^{-1}K(P')^{-1}$, $W_t = P'WP$) satisfy (5.42) with equality. Thus the minimum roundoff noise possible, using $(n+1)^2$ coefficients (in general) and quantization after every non-trivial multiplication can be expressed:

$$(\sigma_0)^2_{opt} = \frac{\Delta_r^2}{12}\left\{(n+1)\left[K_1\right]_{n+1,n+1}\left[W_1\right]_{n+1,n+1} + \frac{n+1}{n}\left(\sum_{i=1}^{n}\mu_i\right)^2\right\} \tag{5.43}$$

assuming we know some $K_1$ and $W_1$ and can solve for the eigenvalues of $KW$.

If in fact we restrict ourselves to the *block optimal* parallel structure [37] with its (fewer) $4n+1$ coefficients, then we are constraining the transformation $P$ to be block diagonal and (5.42) cannot in general be satisfied with equality. However, (5.42) will be true for each *second-order section* ($n=2$). Thus the minimum block optimal product variance can be written:

$$(\sigma_0^2)_{bo} =$$

$$\frac{\Delta_r^2}{12}\left\{(n+1)\left[K_1\right]_{n+1,n+1}\left[W_1\right]_{n+1,n+1} + \frac{3}{2}\left(\mu_1+\mu_2\right)^2 + \frac{3}{2}\left(\mu_3+\mu_3\right)^2 + \cdots\right\} \tag{5.44}$$

This equation in fact suggests a new result — a pairing algorithm for real poles. Once the modes of *KW* are determined, (5.44) will be minimized by pairing modes so that each pair of modes sums to approximately the same quantity as every other pair. In fact, (5.44) may even be lower than (5.43) due to the reduced number of coefficients (noise sources).

The one-level minimum roundoff noise structure developed above can be extended to the case of one-level compensators. Again, we can neglect the A/D noise contribution, which is invariant to structural transformation. Equation (5.33) can be rewritten in terms of its *unscaled* compensator parameters $\Psi_{11}$ and $\Psi_{12}$ as follows:

$$\tilde{Z} = TAT^{-1}\tilde{Z}T^{-1}A'T + \frac{\Delta_r^2}{12}\begin{bmatrix} 0 & 0 \\ 0 & \Lambda_1 \end{bmatrix} \tag{5.45}$$

where

$$T = \begin{bmatrix} I_n & 0 \\ 0 & S_1 \end{bmatrix}$$

and

$$A = \left[\begin{array}{c|c|c} \Phi & 0_n & \Gamma k_{da} \\ \hline \Psi_{12} L k_{ad} & \Psi_{11} & \end{array}\right]$$

By recognizing that the unscaled covariance matrix $Z$ just equals $T^{-1}\tilde{Z}T^{-1}$, we can write (5.45) in a manner similar to (5.37) to produce:

$$Z = AZA' + \frac{\Delta_r^2}{12}\begin{bmatrix} 0 & 0 \\ 0 & \Lambda_1 S_1^{-1} \end{bmatrix} \tag{5.46}$$

The expression for the increase in performance index due to roundoff noise for the scaled system can also be written in terms of the unscaled covariance matrix $Z$: (See (5.34))

$$dJ = \text{trace}\left\{\Upsilon\tilde{Z}\right\} = \text{trace}\left\{\Upsilon T^{-1} Z T^{-1}\right\} \tag{5.47}$$

$$= \text{trace}\left\{T^{-1}\Upsilon T^{-1} Z\right\}$$

$$= \text{trace}\left\{\Upsilon Z\right\}$$

Using an adjoint Lyapunov equation, as in (5.39) and (5.40), we can express (5.46) and (5.47) as follows:

$$W = A'WA + \Upsilon \tag{5.48}$$

$$dJ = \frac{\Delta_r^2}{12}\,\text{trace}\left\{\begin{bmatrix} 0 & 0 \\ 0 & \left\{\Lambda_1 S_1^{-2}\right\} \end{bmatrix} W\right\} \tag{5.49}$$

If we define $W_1$ to be the lower right-hand $(n+1)\times(n+1)$ portion of $W$, then:

$$dJ = \frac{\Delta_r^2}{12}\,\text{trace}\left\{\Lambda_1 S_1^{-2} W_1\right\} \tag{5.50}$$

This expression is identical to the expression in (5.40). From this point on, the derivation of a one-level minimum roundoff noise compensator structure is exactly the same as the Mullis and Roberts and Hwang procedure discussed above (see (5.40)-(5.44)).

Conceptually, the technique described above could be extended to multiple levels. However, the iterative structure optimization procedure considered in Chapter 8 is far more useful for minimizing roundoff noise.

## §5.7 The F8 Example and Compensator Roundoff Noise

This section will examine the roundoff noise and scaling associated with some of the structures discussed in Chapter 3 for an actual sixth-order LQG system. This system is a simplified version of the longitudinal dynamics of the F8 fighter aircraft at flight condition 12 (an altitude of 20,000 feet and a speed of mach 8) [64]. Longitudinal control of the aircraft is restricted to the elevator alone and a single measurement $y$ formed; these simplifications make the plant model single-input single-output, so that all our analysis techniques directly apply. The actual multiple-input multiple-output model could be considered with our techniques, but certain additional issues arise as discussed in Chapter 9.

First-order actuator dynamics are included in the plant model, and and a series integrator is also added. Thus the configuration of figure 5-3 will only have one reference input. Appendix A presents the continuous-time plant model in detail. The sample rate (10 Hertz) is selected to be well above the highest plant pole frequency (12 radians/second). Thus $T$ equals 0.1 seconds. The resulting discrete-time model parameters are also shown in Appendix A.

For this plant model, the design equations of Chapter 2 were followed. The resulting $K$ and $G$ vectors are also given in Appendix 1. All calculations were done in double precision (16 digits, or 64 bits) so that the system parameters and $K$ and $G$ vectors are effectively infinite-precision quantities. The resulting performance index $J$ is 0.00176477. This number is then taken to be the *ideal* value of the performance index, and degradation is measured relative to it.

To five significant digits, the poles and zeros of the (ideal) compensator transfer function are:

| Pole Frequencies | | | Zero Frequencies | | |
|---|---|---|---|---|---|
| $z_{p1}$ | = | 0.29179 | $z_{z1}$ | = | 0.30119 |
| $z_{p2}$ | = | 0.58904 | $z_{z2}$ | = | 0.96728 |
| $z_{p3}$ | = | 0.99514 | $z_{z3}$ | = | 0.99878 |
| $z_{p4}$ | = | 0.99869 | $z_{z4}, z_{z5}$ | = | 0.88189 ± j 0.26766 |
| $z_{p5}, z_{p6}$ | = | 0.73149 ± j 0.40220 | | | |

Figure 5-7: *F8 Compensator Poles and Zeros*

Note that, unlike higher-order digital filters, there are many *real* poles and zeros in this compensator. This fact complicates the pairing issue for parallel and cascade structures. Note also the presence of poles and zeros very near the unit circle at $z=+1$; these singularities can be critical in determining an acceptable structure.

Before discussing the different structures tested, the structure-independent A/D noise contribution will be considered. If we allow a 5% increase in $J$ due to this single noise source, then the procedure outlined in (5.32)-(5.34) using only $\epsilon_{ad}$ results in a 4.98 bit A/D wordlength. (This number does not include the sign bit.) Typically for filtering applications, the A/D wordlength need not be as long as the structure's internal wordlength; the same result appears for this control and estimation application, as will be seen below.

Ten structures were evaluated in terms of their product roundoff noise effects on $J$: the direct form II, five parallel forms including a block optimal structure), three cascade structures, and the simple structure of equation (3.26). The direct form II structure (a) has been described in figure 3-5 and equation (3.20), and has 13 coefficients, including a single scaling multiplier. The first parallel structure (b) is composed of five direct form II sections, one second-order (for the complex pole pair) and four first-order. Each section requires its own scaler, so this structure has a total of 17 coefficients. The next two parallel structures use three second-order sections, and hence the issue of how we pair the four real poles into two sections must be addressed. (There are three different ways.) Parallel structure (c) pairs $z_{p1}$ with $z_{p4}$ and $z_{p2}$ with $z_{p3}$, separating the two near-unit-circle poles, while structure (d) pairs these two poles ($z_{p3}$ and $z_{p4}$), together (see Appendix A). Each structure will require three scalers, for a total of 15 coefficients.

Structures (a) through (d) are all direct form II-based and thus require two precedence levels. Parallel structure (e) is a one-level structure produced by computing $\Psi_\infty = \Psi_2\Psi_1$ where $\Psi_2$ and $\Psi_1$ are from structure (c), and using the

result as a one-level structure (see section 3.3). This structure will still be a parallel combination of second-order sections; each section will be a one-level version of a direct form II section. This structure will have 16 coefficients, one more than (d) or (e). Parallel structure (f) is a minimum roundoff noise block optimal structure as in equation (3.27) and uses the same pole pairing as parallel structures (c) and (e).

As mentioned in Chapter 3, cascade structures involve the issues of *pairing* and *ordering*; in addition to the pairing issues encountered with the parallel structure, the zeros must be paired, and the sections must be ordered. Jackson [61] has described general section ordering and pairing criteria. Consider the $l^{th}$ second-order section:

$$H_l(z) = \frac{1 + a_{l1}z^{-1} + a_{l2}z^{-2}}{1 + b_{l1}z^{-1} + b_{l2}z^{-2}} \tag{5.51}$$

Complex pole pairs and complex zero pairs that are *nearest* each other are placed in the same section (paired). *Nearness* means that we try to pair poles and zeros so as to minimize the peak magnitude ($L_\infty$ norm) of $H_l(z)$ for all $i$. As for section ordering, when direct form II sections are used (with $l_2$ scaling), the noise variance of the filter output *tends* to be minimized by ordering the sections in terms of increasing $\kappa_l$ where

$$\kappa_l = \frac{\|H_l\|_\infty}{\|H_l\|_2} \tag{5.52}$$

(This is not a precise result.) This guideline must be changed if the $L_\infty$ norm of the output is our performance gauge, or if the direct form II section is not used.

Furthermore, Jackson does not consider the pairing of *real* poles. Dehner [65] and Hwang [66] develop general sub-optimal algorithms for selecting good pairing and ordering, but these methods tend to require significant computer time to figure out the ordering and pairing for higher-order filters, and they still do not address the pairing of real poles. Our roundoff analysis will consider just two different cascade pairings/orderings. Cascade structure (g) consists of an arbitrarily-chosen arrangement of poles and zeros (see Appendix A); section 1 contains the complex pole pair ($z_{p5}$,$z_{p6}$) and real zero $z_{z1}$, section 2 contains the near unit-magnitude real poles $z_{p3}$ and $z_{p4}$ and the complex zero pair ($z_{z4}$,$z_{z5}$), and section 3 contains the near unit-magnitude real zeros $z_{z2}$ and $z_{z3}$ with the real poles $z_{p1}$ and $z_{p2}$. Cascade structure (h) splits the near unit-magnitude poles and zeros, and puts the complex pole and zero pairs together in the same section (see Appendix A). Both (g) and (h) require three scalers and a total of 15 coefficients and four precedence levels (see (3.22) and figure 3-6). Cascade structure (I) has the same ordering and pairing as (g), but uses direct form I sections as described in (3.23) and figure 3-7. Hence it has *different* scaling than (g), different scaled coefficients, and fewer scalers.

Finally, the simple structure (j) of (3.26) is treated since this structure (or a one- or two-level version of it) has been often used, even though this structure (scaled) requires an excessive 50 coefficients for the F8 system example.

Appendix A contains the actual modified state space representations of all ten of these ($l_2$-scaled) structures. The ideal values of these coefficients are presented in double precision.

Figure 5-8 summarizes the product roundoff noise results (that is, the noise

| structure | levels | N | wordlength spa | wordlength dpa | max/min $l_2$-scaled coefficient |
|---|---|---|---|---|---|
| (a) direct form II | 2 | 13 | 19.65 | 18.25 | 60050/0.12 |
| (b) parallel direct form II | 2 | 17 | 8.05 | 7.45 | 1.5/0.0046 |
| (c) parallel direct form II | 2 | 15 | 10.18 | 9.39 | 10.5/0.073 |
| (d) parallel direct form II | 2 | 15 | 14.74 | 13.94 | 15.7/0.0015 |
| (e) parallel, 1-level version of (c) | 1 | 16 | 9.78 | 8.99 | 6.3/0.073 |
| (f) block optimal parallel | 1 | 25 | 7.88 | 7.06 | 1.1/0.0029 |
| (g) cascade direct form II | 4 | 15 | 15.69 | 14.68 | 1101/0.00052 |
| (h) cascade direct form II | 4 | 15 | 10.51 | 9.47 | 35/0.073 |
| (i) cascade direct form I | 3 | 14 | 15.52 | 14.36 | 320/0.012 |
| (j) simple | 3 | 50 | 9.01 | 7.54 | 1.6/0.0000003 |

Figure 5-8: *Roundoff Noise Results*

caused by the rounding of multiplier products, not A/D rounding) for these ten structures, assuming optimal $l_2$ scaling and *not* accounting for the finite wordlengths of the coefficients themselves. The 'levels' column lists the number of precedence levels, and the 'N' column lists the number of coefficients including scalers in the structure. The roundoff noise results are presented in terms of the number of signal (wordlength) bits that are required to hold the increase in $J$ due to product roundoff noise to 5% of the ideal value. Again, these numbers do not include the sign bit. Two wordlengths are presented for each structure. The left-hand column (larger) corresponds to the case of roundoff after every nontrivial multiplication and single-precision adders, while the right-hand column corresponds to the case of double-precision adders and quantization after addition. The last column of figure 5-8 shows the maximum and minimum magnitude of the scaled coefficients and is important in determining the coefficient wordlength. The wider the range of values, the more fixed-point coefficient bits will probably be needed to achieve a given level of performance (see Chapter 6).

From figure 5-8 we can see that the different pole pairings associated with

parallel structures (c) and (d) produced results that differed by 4.5 bits. Placing the near-unit magnitude poles in different sections was quite effective. Similarly, of the two cascades (g) and (h), the one with these same two poles in different sections required 5.2 fewer bits. Clearly the pairing/ordering issue is not a trivial question.

Structure (b), the combination of first- and second-order parallel sections, with its 17 coefficients outperformed every other structure except the block optimal. Even so, the extra 8 coefficients of the block optimal structure with second-order sections only gained 0.2 bits of performance over this structure. Thus, when evaluating different structures, it is important to know the block optimal result (for various pairings) so that we can judge whether a suboptimal structure like (b) is effective *enough*. In this case it clearly is. If we are constrained to one level, then (e) is probably best given its 9 fewer coefficients than the optimal and only 1.9 bits poorer performance. Actually, in this case one should check the performance of a one-level version of (b).

As expected from the literature on digital filters, the discrete form II has a very poor noise performance. It is interesting to note also that the simple structure with its many coefficients (and hence many noise sources) performed excellently. It is not clear whether this would be true for the simple structure in general.

The second wordlength column in figure 5-8 shows the gain possible when using double precision adders and fewer quantizers. Depending on the structure tested, a savings of from 0.6 to 1.47 bits was realized. Whether this small savings is enough to justify the higher-precision adders will depend on the particular application.

## §5.8 Summary

Briefly then, we can summarize the major points brought out in this chapter concerning the statistical effects of quantization noise in compensators. The process of scaling a digital feedback compensator requires the consideration of the overall closed-loop control system in which the compensator is embedded. Thus we had to adapt the methods developed for scaling digital filters to this problem. Furthermore, when applying the statistical approach to scaling to the set-point LQG system, we had to consider an alternate configuration for the system. For the analysis of roundoff noise effects in compensators, we again had to adapt the techniques used in digital signal processing to consider the effects of the overall closed-loop system. The development of minimum roundoff noise structures for compensators required similar adaptations. When these methods were applied to a specific control system example, we were able to compare different types of structures in terms of their roundoff noise performance. The importance of the pairing and ordering issue involved with the parallel and cascade structures were shown to be even more complex for compensators, due to the numbers of real poles that are common in control system compensators. Furthermore, the default structure for LQG controllers, the *simple* form, was shown to be a poor choice of structure in general for the LQG compensator.

# Chapter 6: Finite Wordlength Effects: Quantizing the Coefficients

## §6.1 Introduction

The implementation of a discrete-time system described by an ideal infinite-precision transfer function in finite-precision hardware involves several important issues. Chapter 5 has discussed the quantization noise problem, and Chapter 7 will present the issue of limit cycle oscillations. This chapter will consider the problem of quantizing the infinite-precision coefficients of the structure so that they may be stored in a finite-length fixed-point binary representation. As with the roundoff noise question, coefficient quantization effects are also heavily structure-dependent, and thus the analysis of such effects is important when selecting a good structure and its required coefficient wordlength.

Approximating the coefficients of a structure with a finite number of bits will cause a degradation in the system's performance as compared to the ideal. Assuming that a given quantitative performance measure is provided, we can measure the tradeoff in the number of bits versus the degradation. Then, assuming that we specify an acceptable amount of degradation, one must determine the minimum number of coefficient bits needed to meet this goal, and the structure which has the smallest such wordlength.

Whatever the structure, the fewest number of total coefficient bits will be required if we allow each coefficient to have a different wordlength. We certainly will not need fewer total bits after *adding* a constraint such as uniform wordlength. However, the resulting complication in the digital hardware due to non-uniform memory widths and restrictions on the hardware multipliers make this superior apportionment of coefficient bits very costly. For this reason a uniform

fixed-point coefficient wordlength is typically assumed. This assumption will be carried through in the analysis, assuming $n_c$ fractional bits, a sign bit, and enough integer bits to represent the largest coefficient in the structure. We will also assume that each structure has already been scaled, since the scaling operation can radically change the dynamic range of the coefficients, and hence the required wordlength.

The remainder of this chapter is organized as follows. In section 6.2 we will describe different methods for selecting structures that have small required coefficient wordlengths, and different ways of evaluating the required coefficient wordlength once a structure is selected. In particular we will discuss a pole-location-based qualitative method for comparing structures, a direct approach to wordlength evaluation, and a *statistical* approach to structural comparison *and* wordlength determination. We will show that the statistical method has a very important advantage over any other approach — it can be used as the objective function in an iterative structure optimization procedure (see Chapter 8). Sections 6.3 and 6.4 describe the statistical method in detail for the LQG problem, while section 6.5 presents the direct evaluation procedure. Using the F8 system presented in Chapter 5, various coefficient wordlength results and conclusions are presented in section 6.6. Finally, the joint analysis of coefficient wordlength effects and roundoff noise effects is addressed in section 6.7.

## §6.2 Methods of Analysis

Given some measure of performance, there are several methods for calculating the degradation due to coefficient quantization, so that a good structure and the wordlength necessary to meet some allowed degradation level may be

selected. Before discussing these methods, we must address one other important question — how are the ideal coefficients to be quantized? The simplest and most common procedure is to *round* the coefficients to $n_c$ fractional bits. Unfortunately, there is no guarantee that this is the best method in terms of some specific performance metric. In fact, the optimal set of $n_c$-fractional-bit coefficients is usually *not* these rounded values. This fact has given rise to several optimization techniques [67,68,69] for determining the best set of quantized coefficients for a given structure and wordlength. Typically these techniques start *near* the rounded coefficient set (in discrete coefficient space) and search for minima. Unfortunately, these methods can be extremely time-consuming, with the resulting coefficient set not necessarily that much better than that obtained by rounding. Consequently, we will assume that finite-wordlength coefficients are produced by rounding the ideal values.

The effect of a quantized coefficient on any performance measure is essentially a *sensitivity* question. From a frequency-domain viewpoint, having coefficients of finite wordlength implies that there are only a finite number of possible pole and zero locations in the $z$-plane. Thus one approach to the selection of a structure with minimal coefficient quantization effects could be accomplished by examining a graph, or grid, of these locations; the coefficient sensitivity in an area of high grid density would be small. Thus, the structure which had the densest grid in the area of the desired poles and zeros would be chosen. Several structures have been described in terms of pole location grids; for example, the *coupled form* second-order section of Rader and Gold [70] has a uniform square grid over the entire $z$-plane, while the direct form II has a non-uniform grid, densest near $z = \pm j$. Avenhaus [34], Abu-El-Haija, Shenoi, and Peterson [71], and

Agarwal and Burrus [72] have described second-order sections whose grids are densest near $z = +1$, thus making them excellent for implementing lowpass filters. Avenhaus has also presented other sections and their respective pole location grids. Such a general approach to filter structure selection at least has an intuitive appeal. Of course, there is no guarantee that a structure with high grid density for the desired pole locations will necessarily be the *best* structure in terms of any other measure of performance degradation due to coefficient quantization effects, especially when using performance measures such as the trace of the error covariance (for a Kalman filter), or the performance index $J$ (for LQG systems), or phase margin (for a classical control system).

Given any set of quantized coefficients, the most direct and accurate way to evaluate the effect of finite wordlength on performance would be to recompute *for the quantized coefficient values* the entire transfer function, performance index $J$, phase margin, or whatever quantitative measure is appropriate. In fact, this is the approach taken by Sripad [13] for analyzing the effects of finite wordlength coefficients. While this method has the virtue of being accurate, it tells us only one point on the performance/wordlength tradeoff curve. The performance measure would have to be reevaluated for each potential wordlength until the desired degradation level has been bracketed (bounded above and below) by wordlengths differing only by one bit. Then the larger of the two wordlengths would be the required coefficient wordlength for that structure. Such a brute-force approach could be quite time-consuming, especially when we wish to compute the required number of bits for several candidate structures.

What would be quite convenient would be to have a procedure where a single evaluation established the behavior of the performance/wordlength tradeoff

curve. The required wordlength could then be estimated easily from knowing the allowed degradation level. Also, since the wordlength *must* be integral, some accuracy can be sacrificed to gain simplicity, as long as the required wordlength is not *underestimated*. More importantly, if the coefficient wordlength estimate is continuous in nature, that is, not confined to an integral number of bits, then it is possible to apply an optimization technique [17] to synthesize better structures. In this procedure, which we will describe in Chapter 8, continuous transformations are applied to an initial structure. These transformations are determined by a gradient search technique based on some continuous, differentiable scalar objective function of the coefficients of the structure. Certainly, if our required wordlength is strictly integral, it is not differentiable.

The concept of a *statistical* estimate of wordlength has both the advantages mentioned above. This approach originated in the study of digital filters with the work of Knowles and Olcayto [73]. Avenhaus [67] applied this idea to the digital filter power transfer function (as a performance measure), and later Crochiere [32,74] used the concept with the filter transfer function magnitude and a wordlength-optimization procedure.

The remainder of this section will review the basic development of the statistical wordlength measure for digital filters [74]. Consider a general scalar measure of performance $f$ that is a function of a set of coefficients, and is continuous and differentiable. For example, the error in the transfer function magnitude at a specific frequency, the integrated squared error in the transfer function magnitude, and the performance index for an infinite-time-horizon LQG problem are acceptable measures. With a finite-precision implementation, the resulting $f$ will depend on the $N$ quantized coefficients $(c_1, c_2, \cdots c_N)$ of the structure. The

value of $f$ associated with any particular finite-precision structure will reflect a degradation in performance as compared to the ideal (infinite-precision) value $f_\infty$. Assume that this degradation $df$ can be expanded in a Taylor's series about the ideal value. Keeping only first-order terms,

$$df(c_1, c_2, \cdots c_N) \approx \sum_{i=1}^{N} \left( \frac{\partial f}{\partial c_i}\bigg|_\infty dc_i \right) \tag{6.1}$$

where $c_i$ is the $i^{th}$ coefficient to be rounded, $dc_i$ is the error due to rounding, and $\frac{\partial f}{\partial c_i}\Big|_\infty$ is the first partial derivative of $f$ evaluated at the unrounded coefficient values. Note that coefficients such as 3, 2, 1, and ½ are normally not affected by rounding and should not be included in the sum (6.1).

If $\Delta$ is the quantization step size $2^{-n_c}$, the fraction represented by the least significant bit of the fixed-point coefficient word, then each $dc_i$ must lie between $\pm\frac{\Delta}{2}$. Given the partial derivatives in (6.1), we could then upper bound the error $df$, producing a *very pessimistic* wordlength estimate:

$$df < \frac{\Delta}{2} \sum_{i=1}^{N} \left| \left( \frac{\partial f}{\partial c_i}\bigg|_\infty \right) \right| \tag{6.2}$$

The basic idea behind statistical wordlength is to treat an ensemble of structures. Over this ensemble, the coefficient errors $dc_i$ can be thought of as uniformly-distributed zero-mean *uncorrelated* random variables, each of variance $\frac{\Delta^2}{12}$. The error $df$ is therefore also zero-mean with a variance:

$$(\sigma_{df})^2 = \frac{\Delta^2}{12} \sum_{i=1}^{N} \left( \left. \frac{\partial f}{\partial c_i} \right|_{\infty} \right)^2 \tag{6.3}$$

For large $N$, the central limit theorem can be applied to justify a Gaussian distribution for $df$ . Thus with a given probability, say 95%, one can determine the variance needed for the error $df$ to remain within some prescribed bound. In other words 95 out of 100 of the structures in the ensemble will result in systems where $df$ remains within this bound.

From a table of the Gaussian distribution,

$$\Pr\left[ \left| df \right| \leq 2\sigma_{df} \right] = 0.954 \tag{6.4}$$

If the quantity of interest $f$ is constrained to lie within $\pm E_0$ (the degradation level) of the ideal $f_\infty$, then (6.4) implies that $\sigma_{df}$ equal $\frac{E_0}{2}$. This result can be combined with (6.3) to produce an estimate of the parameter $\Delta$:

$$\Delta = \frac{\sqrt{3} E_0}{\left\{ \sum_{i=1}^{N} \left( \left. \frac{\partial f}{\partial c_i} \right|_{\infty} \right)^2 \right\}^{\frac{1}{2}}} \tag{6.5}$$

Given $\Delta$, the statistical wordlength can be defined to be:

$$SWL = I + \log_2 \frac{1}{\Delta} \tag{6.6}$$

The first term in (6.6) represents the number of bits necessary to represent the integer portion of the coefficient word (bits to the left of the fixed binary point) and the second term gives the number of bits $n_c$ necessary for the

fractional portion of the coefficient word (bits to the right of the binary point). The sign bit is not included in this expression.

In the digital filter area, Crochiere [31,32,74] presents a number of results comparing the statistical wordlength of structures using the transfer function magnitude as the performance measure $f$. Since this choice of $f$ is frequency-dependent, the resulting estimate is also frequency-dependent. The final wordlength can be selected as the maximum of the estimates over the frequency range of interest. In the examples treated by Crochiere, the statistical wordlength estimate was 1 to 3 bits conservative as compared to the actual minimum number of bits necessary to just meet the transfer function error limit. In a related work by Chan and Rabiner [75], which considered a large number of finite impulse response filters and a similar statistical approach to coefficient wordlength, the resulting 95% confidence level estimates were also observed to be conservative. Crochiere [32,74] was also able to use statistical wordlength as the basis for a filter optimization procedure quite different from the technique we will present in Chapter 8 (but not applicable to LQG compensators).

## §6.3 Statistical Wordlength and LQG Systems

As mentioned in Chapters 1 and 2, it is natural to use the performance index $J$ of (2.3) as the measure of performance $f$ for a steady-state LQG system. Using the approach of the previous section, the change in $J$ would be estimated by:

$$dJ(c_1,c_2,\ldots c_N) \approx \sum_{i=1}^{N} \left( \frac{\partial J}{\partial c_i}\bigg|_{\infty} dc_i \right) \tag{6.7}$$

However, the optimal nature of the LQG control problem forces all the first-order sensitivities $\frac{\partial J}{\partial c_l}$ to be zero. Therefore a higher-order approximation is necessary:

$$dJ \approx \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left( \frac{\partial^2 J}{\partial c_i \partial c_j} \bigg|_{\infty} dc_i \, dc_j \right) \tag{6.8}$$

The use of second-order terms (not used in digital filter analysis) is a unique aspect of our statistical wordlength formulation. However, the use of these terms would be implicit in any statistical estimate based on the error in an *optimized* scalar performance measure. If a digital filter was designed by minimizing the integrated squared error between the desired and actual filter transfer function magnitude characteristic, then a statistical wordlength estimate based on this performance measure would have to use second-order sensitivities — all first-order sensitivities would be zero. Thus our statistical wordlength derivation could be an extension to the techniques of digital signal processing. However, when the overall filter statistical estimate is taken to be the maximum over a set of estimates made at specific frequencies (each based on the transfer function magnitude error at that frequency), then of course the first-order sensitivities for each of those estimates would be non-zero no matter how the original filter was designed. This was the case considered by Crochiere. Frequently in fact, digital filters are not designed by minimizing a differentiable scalar criterion. Thus one would have to use the approach taken by Crochiere for developing a statistical wordlength estimate.

Proceeding from (6.8), recall that all the errors $dc_i$ and $dc_j$ are assumed

to be uncorrelated for $i \neq j$. Thus, the mean of $dJ$ will no longer be zero:

$$E(dJ) = \frac{1}{2} \sum_{i=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i^2} \right|_{\infty} \right) E\left[ (dc_i)^2 \right] \tag{6.9}$$

For convenience, define the random variable $\epsilon$ to be the square of $dc_i$. Its mean and variance can be shown to be $E(\epsilon) = \bar{\epsilon} = \frac{\Delta^2}{12}$ and $E(\epsilon^2) = \overline{\epsilon^2} = \frac{\Delta^4}{180}$. The second moment and variance of $dJ$ can be written as follows:

$$E\left[ (dJ)^2 \right] = \frac{\overline{\epsilon^2}}{4} \sum_{i=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i^2} \right|_{\infty} \right)^2 + \frac{\left(\bar{\epsilon}\right)^2}{4} \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i^2} \right|_{\infty} \right) \left( \left. \frac{\partial^2 J}{\partial c_j^2} \right|_{\infty} \right)$$

$$+ \frac{\left(\bar{\epsilon}\right)^2}{2} \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i \partial c_j} \right|_{\infty} \right)^2 \tag{6.10}$$

$$(\sigma_{dj})^2 = \frac{\sigma_\epsilon^2}{4} \sum_{i=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i^2} \right|_{\infty} \right)^2 + \left(\bar{\epsilon}\right)^2 \sum_{\substack{i=1 \\ i > j}}^{N} \sum_{j=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i \partial c_j} \right|_{\infty} \right)^2 \tag{6.11}$$

Recall the application of the central limit theorem in section 6.2. We can make the same assumption for our higher-order statistical wordlength derivation. For the usual digital filtering estimate, the coefficient quantization could either decrease or increase the error in the transfer function magnitude at any specific frequency. This error was zero-mean. In the control case, the value of $J$ can only *increase* under coefficient quantization. Thus we need only have a specification on the maximum allowed value of J including the degradation due to

coefficient quantization: $J_\infty + E_0$. Following the general approach of section 6.2, we must relate this value to the two-sigma point in the distribution for $dJ$ (See figure 6-1):

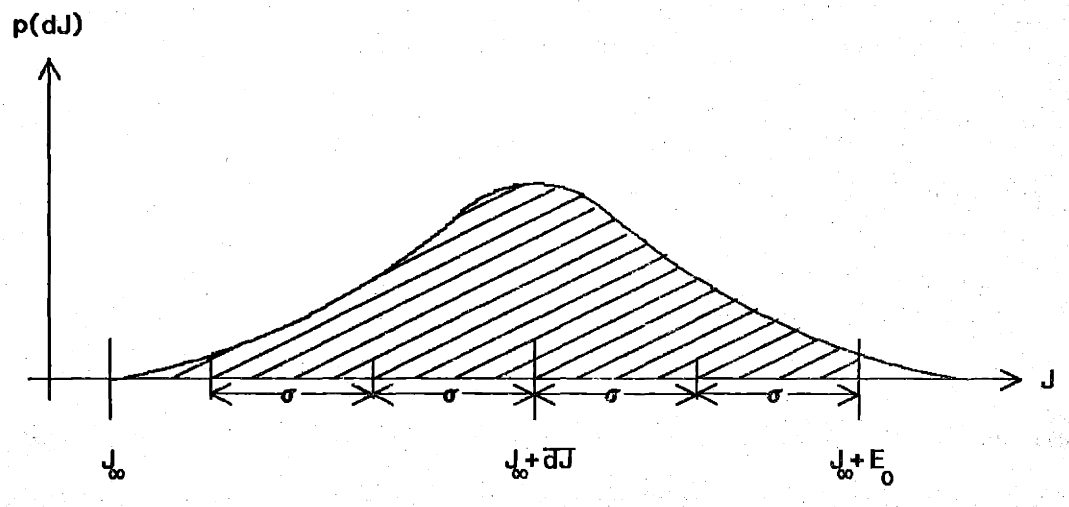



Figure 6-1: Probability Density of dJ

$$J_\infty + E_0 = J_\infty + \overline{dJ} + 2\sigma_{dJ} \tag{6.12}$$

This choice of $\sigma_{dJ}$ gives a 97.5% confidence level in terms of remaining below the allowed deviation $E_0$. Combining (6.11), (6.12), and the values of $\bar{\epsilon}$ and $\sigma_\epsilon$, we can derive an expression for $\Delta^2$:

$$\frac{1}{\Delta^2} = \frac{1}{24E_0} \sum_{i=1}^{N} \left( \frac{\partial^2 J}{\partial c_i^2} \bigg|_\infty \right) +$$

$$\frac{1}{6E_0}\left[\sum_{\substack{i=1\\ i>j}}^{N}\sum_{j=1}^{N}\left(\frac{\partial^2 J}{\partial c_i \partial c_j}\bigg|_{\infty}\right)^2 + \frac{1}{5}\sum_{i=1}^{N}\left(\frac{\partial^2 J}{\partial c_i^2}\bigg|_{\infty}\right)^2\right]^{\frac{1}{2}} \tag{6.13}$$

Using (6.6), the SWL can be written:

$$SWL = I + \frac{1}{2}\log_2\left(\frac{1}{\Delta^2}\right) \tag{6.14}$$

There are several important distinctions between the statistical wordlength method as described in section 6.2 and the expression (6.14). First, the requirement of second derivative terms has led to a fairly complex expression for the SWL — an efficient computational procedure will be critical. Second, since the performance index $J$ is not frequency dependent, neither is the statistical wordlength estimate. Only one evaluation will be needed, rather than one per frequency as with the transfer function-based filter wordlength estimate. Another distinction involves the Gaussian assumption. The analysis in (6.12)-(6.14) applied the central limit theorem to justify this distribution. Yet we know that the distribution of $dJ$ must be one-sided ($dJ$ must be positive). Thus the 97.5% probability figure may not be as accurate as the digital filter probability of 95%. However, it is not really important whether 95 out of 100 structures have statistical estimates that are conservative, or 85 out of 100, and so forth.

The final distinction between the usual filtering estimate described in section 6.2 and the LQG controller estimate is the non-zero mean degradation $\overline{dJ}$. For the filtering case, the mean degradation in the transfer function is zero. For the LQG development, it is possible to form an estimate without taking into account the standard deviation of the error $dJ$. If we set the mean degradation value to

equal the allowed degradation $E_0$, then using (6.9):

$$\overline{dJ} = E_0 = \frac{\Delta^2}{24} \sum_{i=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i^2} \right|_{\infty} \right) \tag{6.15}$$

From (6.15), we can write an expression for $\Delta^2$:

$$\Delta^2 = \frac{24E_0}{\sum_{i=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i^2} \right|_{\infty} \right)} \tag{6.16}$$

A *mean* statistical wordlength (*MSWL*) can now be defined, using (6.14) and (6.16):

$$MSWL = I + \frac{1}{2}\log_2 \left( \frac{\sum_{i=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_i^2} \right|_{\infty} \right)}{24E_0} \right) \tag{6.17}$$

The interpretation of this *MSWL* estimate is that one half of the structures using this wordlength will have more degradation than $E_0$, and one half will have less. *Whether or not the MSWL is a useful estimate* depends on the width of the *dJ* distribution in figure 6-1 and on the change in this distribution from structure to structure. In other words, it will depend on the tightness in the relationship between the *SWL* and the *MSWL* estimates. Consequently, we will compute both estimates for a selection of structures. The advantage of the *MSWL* is clear — reduced complexity and hopefully significantly less time for its evaluation.

At this point it is convenient to mention the analysis of sub-optimal compen-

sators. If the sub-optimal compensator results from a parameter optimization problem [19,20], then the first-order sensitivities will all be zero and the statistical wordlength approach developed in this chapter can be used. If the sub-optimal design is only an approximation of an optimal design, then we can still apply this method. The only difference would be the inclusion of first-derivative $\frac{\partial J}{\partial c_l}$ terms (first-order sensitivities); these terms would be non-zero since the compensator is not optimal, or even locally optimal as in the parameter optimization designs.

Again, it is important to realize that the estimates derived in this section for LQG compensator coefficient wordlength could also be useful for digital filters, as long as the filter design optimizes some scaler differentiable objective function.

## §6.4 Computing the Statistical Wordlength

As in Chapter 5, the trace form of $J$ will be convenient for computing statistical wordlength. Recall the following two equations from Chapter 5:

$$J = \text{trace}\,(\Upsilon\, Z) \tag{6.18}$$

$$Z = AZA' + C \tag{6.19}$$

where $\Upsilon$ contains the weighting matrices $Q$, $M$, and $R$ as in (5.34), and $A$ and $C$ are defined by:

$$A = \left[\begin{array}{c|c|c} \Phi & 0_n & \Gamma k_{da} \\ \hline \Psi_{12} L k_{ad} & \Psi_{11} & \end{array}\right]$$

and

$$C = \begin{bmatrix} \Theta_1 & 0 \\ 0 & \left(\Psi_{12}\Theta_2\Psi_{12}'\right) \end{bmatrix}$$

Assume that the structure has *already* been scaled, so that $\Psi_{11}$ and $\Psi_{12}$ contain the infinite-precision scaled compensator parameters. If (6.18) is evaluated with these infinite-precision parameters, the resulting value of the performance index $J_\infty$ will be independent of the structure chosen. However, the partial derivatives of $J_\infty$ with respect to the coefficients of the structure, evaluated at the ideal coefficients values, *will* of course be structure-dependent. The second-partial derivative of (6.18) can be written:

$$\frac{\partial^2 J}{\partial c_i \partial c_j} = \text{trace } \Upsilon \frac{\partial^2 Z}{\partial c_i \partial c_j} \tag{6.20}$$

Thus the partials of $Z$ (each a matrix) must be computed efficiently. Taking the first derivative with respect to $c_i$ of (6.19) produces:

$$\frac{\partial Z}{\partial c_i} = A\frac{\partial Z}{\partial c_i}A' + Q_i + Q_i' \tag{6.21}$$

where

$$Q_i = \frac{\partial A}{\partial c_i}ZA' + \begin{bmatrix} 0 & 0 \\ 0 & \left\{\frac{\partial \Psi_{12}}{\partial c_i}\Theta_2\Psi_{12}'\right\} \end{bmatrix}$$

The second partials of $Z$ can now be written:

$$\frac{\partial^2 Z}{\partial c_i \partial c_j} = A \frac{\partial^2 Z}{\partial c_i \partial c_j} A' + X_{ij} + X_{ij}' \tag{6.22}$$

where

$$X_{ij} = \frac{\partial A}{\partial c_j} \frac{\partial Z}{\partial c_i} A' + \frac{\partial A}{\partial c_i} \frac{\partial Z}{\partial c_j} A' + \frac{\partial A}{\partial c_i} Z \frac{\partial A'}{\partial c_j} + \frac{\partial^2 A}{\partial c_i \partial c_j} Z A'$$

$$+ \begin{bmatrix} 0 & 0 \\ 0 & \left\{ \frac{\partial \Psi_{12}}{\partial c_i} \Theta_2 \frac{\partial \Psi_{12}'}{\partial c_j} + \frac{\partial^2 \Psi_{12}}{\partial c_i \partial c_j} \Theta_2 \Psi_{12}' \right\} \end{bmatrix}$$

Rather than solving (6.22) $N^2$ times (extremely time-consuming) we can apply the adjoint method used in Chapter 5. Equations (6.20) and (6.22) can be replaced by the following two equations:

$$\frac{\partial^2 J}{\partial c_i \partial c_j} = \text{trace}\, \hat{U} \left( X_{ij} + X_{ij}' \right) = 2\, \text{trace} \left( \hat{U} X_{ij} \right) \tag{6.23}$$

$$\hat{U} = A' \hat{U} A + \mathbb{T} \tag{6.24}$$

where $\hat{U}$, $A$, and $\mathbb{T}$ are all $(2n+1)\times(2n+1)$ matrices, and $A$ and $\mathbb{T}$ can be found in (6.19) and (5.34).

Further simplification is possible when evaluating (6.23) once $\hat{U}$ is computed. The matrices $A$ and $\Psi_{12}\Theta_2\Psi_{12}'$ can be expressed in terms of $\Psi_\infty$:

$$A = \begin{bmatrix} I_n & 0 \\ 0 & \Psi_\infty \end{bmatrix} \left[\begin{array}{c|c|c} \Phi & 0_n & \Gamma k_{da} \\ \hline 0 & I_{n+1} & \\ \hline Lk_{ad} & 0 & \end{array}\right] \tag{6.25}$$

$$\Psi_{12}\theta_2\Psi_{12}' = \begin{bmatrix} I_n & 0 \\ 0 & \Psi_\infty \end{bmatrix} \begin{bmatrix} 0_n & 0 & 0 \\ 0 & 0_{n+1} & 0 \\ 0 & 0 & \theta_2 \end{bmatrix} \begin{bmatrix} I_n & 0 \\ 0 & \Psi_\infty' \end{bmatrix} \tag{6.26}$$

Thus the expression for $X_{ij}$ can be grouped into four terms based on derivative quantities; one involving $\frac{\partial \Psi_\infty}{\partial c_j}$ and $\frac{\partial Z}{\partial c_i}$, one involving $\frac{\partial \Psi_\infty}{\partial c_i}$ and $\frac{\partial Z}{\partial c_j}$, a third involving $\frac{\partial^2 \Psi_\infty}{\partial c_i \partial c_j}$, and the last involving $\frac{\partial \Psi_\infty}{\partial c_i}$ and $\frac{\partial \Psi_\infty}{\partial c_j}$. The first partials of $Z$ are known from solving (6.21) $N$ times. Now let us examine the derivatives of $\Psi_\infty$. Since a coefficient can occur once in only one $\Psi_i$ matrix, the first partial of $\Psi_\infty$ with respect to $c_i$ (assume $c_i$ is located at index $(k,l)$ in $\Psi_m$) will be:

$$\frac{\partial \Psi_\infty}{\partial c_i} = \Psi_q \Psi_{q-1} \cdots \Psi_{m+1} E_{kl} \Psi_{m-1} \cdots \Psi_i \tag{6.27}$$

where $E_{kl}$ is defined to be a *unit element matrix* of the same dimensions as $\Psi_m$. This matrix is all zero except for a single unity entry at index $(k,l)$. Similarly, if $c_j$ is located in $\Psi_t$ at index $(r,s)$, we can write:

$$\frac{\partial \Psi_\infty}{\partial c_i \partial c_j} = \Psi_q \cdots \Psi_{m+1} E_{kl} \Psi_{m-1} \cdots \Psi_{t+1} E_{rs} \Psi_{t-1} \cdots \Psi_1 \qquad (6.28)$$

We can infer from (6.28) that if $c_i$ and $c_j$ are in the same matrix $\Psi_m$ (thus $m=t$), then $\frac{\partial^2 \Psi_\infty}{\partial c_i^2}$ must be zero. This fact simplifies the calculation of $X_{ij}$ to some extent (significant for the *MSWL* estimate, which only requires $X_{ij}$ for $i=j$). Appendix C presents further details regarding the evaluation of (6.23).

Unfortunately, the evaluation of (6.23) still requires the computation of equation (6.21) for all $N$ coefficients. However, these computations can also be simplified. The Lyapunov solution method used in this analysis is that of Barraud [76], and has several distinct parts. Given an equation like (6.19), this method will:

(1) Compute an orthogonal transformation matrix $P$ that converts $A$ to *upper Schur form* (upper triangular except for the first sub-diagonal row): $A_S = P'AP$

(2) Use $P$ to transform $C$ to $C_S$: $C_S = P'AP$

(3) Solve the transformed equation $Z_S = A_S Z_S A_S' + C_S$ by a back substitution technique

(4) Transform the result $Z_S$ to $Z$ via $Z = PZ_S P'$

The number of operations involved in each step is proportional to $(2n+1)^3$ if $Z$, $A$, and $C$ are $(2n+1)\times(2n+1)$. However, by far the majority of the computations are involved in step 1, which performs an eigenvalue-eigenvector analysis of $A$. Step 3 requires (approximately) 5% to 10% of the total time, depending on the particu-

lar $A$ matrix. The important point to realize is that step 1 need only be performed once for all $N$ equations (6.21). In fact steps 2 and 4 can also be simplified by including the $P$ and $P'$ multiplications in the matrices $M1$ and $M2$ described above for the $X_{ij}$ terms. Using this method, there will still be a proportionality to $N(2n+1)^3$ in computing (6.21), but it will be many times smaller than for the full four-step procedure.

In summary, the computational procedure for statistical wordlength primarily involves the second derivatives of $J$ required for (6.10). Assuming that computation time is dominated by the number of multiplies, the following approximate dependence of the computation time on the number of coefficients $N$ and the (augmented) system order $2n+1$ exists:

$$t_{SWL} \propto N^2(2n+1)^2 + N(2n+1)^3 + (2n+1)^3$$

For the *MSWL* estimate, this proportionality will be reduced:

$$t_{MSWL} \propto N(2n+1)^3 + (2n+1)^3$$

Thus, as $N$ increases, the *MSWL* estimate becomes computationally more and more efficient as compared with the *SWL* estimate.

## §6.5 Direct Wordlength Computation

For comparison, it is important to include the direct method for determining the coefficient wordlength required to meet or exceed the degradation level $E_0$. Basically, this procedure will involve selecting a test wordlength, rounding the coefficients to that wordlength, and then forming the (finite-precision) matrices $\Psi_i$, $A$, and $C$. Using these finite-precision parameters, the Lyapunov equation (6.19)

must be solved and the trace (6.18) evaluated. The resulting value of $J$ can be compared to $J_\infty + E_0$, and then a decision made whether to alter the test wordlength up or down.

If the performance index were strictly monotonic in the coefficient wordlength, then a binary search algorithm could be designed that would always succeed in finding the required wordlength. For example, starting at some large initial test wordlength, one could decrement the test wordlength 10 bits at a time until the performance index exceeded the value $J_\infty + E_0$, then increment the test wordlength in smaller steps until the performance index was below $J_\infty + E_0$, and so forth. However, $J$ need not be strictly monotonic in wordlength, since the coefficient rounding operation is so nonlinear. However, $J$ *is* roughly monotonic. Thus, the search procedure must try to account for possible anomalies in the behavior of $J$. One other pitfall must be avoided; if the test wordlength is so small that the resulting feedback system is *unstable*, then the computed $J$ value will be meaningless. One simple way to test for this possibility would be to examine the resulting eigenvalues, which are a by-product of the Lyapunov solution method of Barraud.

The method we have used is based on the above discussion. After loosely bracketing the allowed degradation $E_0$ with two test wordlengths, the lower of which is tested to guarantee stability, an exponential curve is fit to these two points. Using $E_0$ and this curve, a reasonable choice of a new test wordlength can be made. From this point, the test wordlength is stepped a bit at a time until the required wordlength is established. The details of the algorithm are shown below:

(1) Bracket the wordlength $w$ with the initial values $wmax$=48 and $wmin$=0. Initialize the increment $i$ at 10, and set the initial value of $w$ near the value $wmax$. Compute the ideal $J_\infty$ using the double-precision coefficient values, and add an allowed level of degradation to produce the desired performance $J_0$.

(2) Decrement the wordlength $w$ by $i$.

(3) Test for a negative wordlength $w$. If found, set $w$ to 1.

(4) Round the ideal coefficient values to wordlength $w$, and compute the resulting test value $J_t$ of the performance index.

(5) Test for instability by comparing $J_t$ to $J_\infty$. If $J_t$ is smaller, the system with coefficients of wordlength $w$ is unstable. Then increment $w$ by $i$, halve the increment size, and return to step (2). Otherwise, if $J_t$ is larger than $J_\infty$, continue.

(6) Test to see if $J_t$ is between $J_\infty$ and $J_0$. If so, set $wmax$ to the current value of $w$ and return to step (2). Otherwise, set $wmin$ to the current value of $w$ and continue. Thus we have bracketed the required wordlength with $wmax$ and $wmin$, and know the performance levels for each of these wordlengths.

(7) Using the two wordlength/performance points found in step (6), and also the ideal performance value $J_\infty$ (associated with some very large wordlength, say 100), fit an exponential curve to describe the performance index as a function of the wordlength. Interpolate to find a next guess at the required wordlength. Round the coefficients to this wordlength and compute the resulting performance index.

(8) If this value is greater than $J_0$, increase $w$ a bit at a time until the resulting performance level is below $J_0$. The corresponding $w$ will be the required wordlength. If however the performance level from step (7) is below $J_0$, decrease $w$ a bit at a time until the resulting performance level is above $J_0$. The corresponding wordlength $w$, *plus one*, will be the required wordlength.

The direct algorithm may be time-consuming as compares to the statistical method because it requires repeated solutions of the Lyapunov equation (6.19) until we bracket the desired performance, and no simplifications are possible from one solution to the next since each finite-precision $A$ matrix is different. If an average of $n_i$ iterations are required to establish the required wordlength, then the dominant number of multiply operations required to compute this *true wordlength* (*TWL*) is proportional to:

$$t_{TWL} \propto n_i(2n+1)^3$$

Thus, a comparison between the statistical estimates *SWL* and *MSWL* and the *TWL* described above will depend upon $n$, $n_i$, $N$, and the constants of proportionality. However, as the number of coefficients increases, the statistical estimates will become less and less efficient, while the true wordlength computation time remains essentially constant. Recall though, that the statistical estimate is still useful as the basis for a wordlength optimization procedure as discussed in Chapter 8. The true wordlength method could not be used for such a procedure, since it is not continuous and thus not differentiable.

## §6.6 The F8 System and the Coefficient Wordlength Issue

The effects of finite coefficient wordlength were evaluated for the F8 system example and the ten structures described in Chapter 5. The results are presented in figure 6-2 using the following format. Column 1 lists the number of integer bits required for the coefficient word; this value is obtained from the largest scaled coefficient value (see figure 5-8). The next three columns list the statistical estimates *SWL* and *MSWL*, and finally the true wordlength as evaluated in section 6.5. In each case, the execution time in seconds for each wordlength determination method is listed in parenthesis following each entry. These times are subject to some small amount of uncertainty depending on specific run-time conditions, so they must be regarded as approximate. Again, the wordlengths listed represent the number of coefficient bits (not including the sign) required to achieve at most a 5% increase in the performance index *J*. Finally, the last column of figure 6-2 lists the number of bits by which the *SWL* estimate exceeds the actual required wordlength.

| structure | | I | SWL | MSWL | TWL | SWL-TWL |
|---|---|---|---|---|---|---|
| a) | direct form II | 16 | 35.99 (0.81) | 35.05 (0.70) | 32 (1.2) | 3.99 |
| b) | parallel d.f. II | 1 | 6.84 (0.93) | 6.16 (0.78) | 6 (1.08) | 0.84 |
| c) | parallel d.f. II | 4 | 12.38 (0.87) | 11.52 (0.78) | 11 (1.26) | 1.38 |
| d) | parallel d.f. II | 4 | 19.02 (0.85) | 18.14 (0.77) | 13 (1.08) | 6.02 |
| e) | 1-level from (c) | 3 | 11.08 (0.90) | 10.22 (0.78) | 10 (1.19) | 1.08 |
| f) | block optimal | 1 | 7.02 (1.26) | 6.2 (0.91) | 7 (1.11) | 0.02 |
| g) | cascade, d.f. II | 11 | 26.25 (0.83) | 25.38 (0.72) | 21 (1.21) | 5.25 |
| h) | cascade, d.f. II | 6 | 14.61 (0.86) | 13.81 (0.72) | 14 (1.36) | 0.61 |
| i) | cascade, d.f. I | 9 | 24.25 (0.84) | 23.38 (0.71) | 20 (1.1) | 4.25 |
| j) | simple | 1 | 9.05 (2.44) | 8.25 (1.29) | 9 (1.71) | 0.05 |

Figure 6-2: *F8 Coefficient Wordlength Results*

A great deal of information may be drawn from figure 6-2. First, we can discuss the performance of the ten structures with regard to coefficient wordlength. Referring to the *TWL* values, we can see that the parallel structure (b) using first- and second-order direct form II sections, and the block optimal parallel structure (f) performed the best, needing only 6 and 7 bits respectively. Quite acceptable performance was also achieved with the simple structure (j) (9 bits), the one-level parallel structure (e) (10 bits), and the parallel structure (c) (11 bits). As with the roundoff noise results of Chapter 5, the direct form II structure (a) performed the worst. For the two parallel and two cascade structures using all second-order sections but with two different pole pairings, the pairing that was better for roundoff noise ((c) and (h)) was also superior for coefficient sensitivity — 2 bits better for the parallel case, and 7 bits better for the cascade.

In fact, if we rank the structures on the basis of their required coefficient wordlengths (b, f, j, e, c, d, h, i, g, a) and then also on the basis of their signal variable wordlengths/roundoff noise performance (f, b, j, e, c, h, d, i, g, a), we can

see a very strong correlation. The orderings are nearly identical — only the adjacent structures *b* and *f* are interchanged, as are *h* and *d*. The correlation between good roundoff noise performance and low coefficient sensitivity has been well-publicized for digital filter structures [17,40,48,77,78]. Of course, these results pertain to the sensitivity of the transfer function *magnitude* to its coefficients. From our results, this correlation seems to carry directly over to the control compensator setting.

One point to be cognizant of is that certain coefficients in a structure, when rounded to the *TWL* wordlength, may in fact become zero or unity, thus eliminating them as multipliers. This situation occurs in the simple structure (j), reducing the number of coefficients from 50 to 40, in the block optimal structure (f), reducing the number of coefficients from 25 to 24, and in the parallel structure (b), reducing the number of coefficients from 17 to 16. Such reduction should factor into the structure selection procedure.

Taking the number of multiplies, number of precedence levels, roundoff noise performance, and required coefficient wordlength all into account, parallel structure (b), which uses first-order sections for real poles and second-order sections for complex poles, is probably the best choice. To achieve an *overall* 3% maximum increase in $J$ with this structure, we could use an 8-bit A/D converter, 8-bit coefficients, and 10-bit signal variables. (Due to the quadratic nature of $J$, each extra bit reduces the increase in $J$ by approximately a factor of four.) Each of these wordlengths includes the sign bit. If circumstances *required* a one-level structure for a short sampling period $T$, then we would probably use the block optimal structure and 24 hardware multipliers. Any final decision as to structure selection is of course application-dependent.

The above discussion applies to the actual wordlengths found by the direct method. Now let us examine how useful it would be to make the comparison of structures using the *SWL* statistical estimate. For the ten structures shown, the *SWL* estimate ranged from 0 to 6 bits conservative, which is quite a wide range. However, this situation is easily explained. Structures (d), (g), and (i) had the poorest estimates. Not coincidentally, all three of these structures have two particular coefficients in common, -.9938344 and 1.9938281 (see Appendix A), and these two coefficients dominate in the expression for statistical coefficient wordlength for these examples. Removal of these two coefficients from the statistical wordlength analysis produces estimates within *one* bit of the true wordlength. Thus these cases represent low probability events (from the left-hand tail of the distribution in figure 6-1). In any case, these particular two coefficients resulted from pairing the two real near unit-magnitude poles, which has already been shown to be a poor choice with respect to finite wordlength performance. Of the ten structures, the *SWL* estimate is excellent (0 to 1.1 bits conservative) for the five lowest coefficient wordlength structures and the cascade (h).

As for a comparison between the *SWL* and *MSWL* estimates, the *MSWL* value was consistently .68-.94 bits below the *SWL* value. This tight range of values suggests that the distribution of $dJ$ shown in figure 6-1 is quite narrow. Thus the *MSWL*, which is simpler to compute, may well be preferable to the *SWL*. One could compute the *MSWL* and then add some fixed number, say one bit, for an estimate. The primary advantage to using the *MSWL* estimate over the *SWL*, given their apparent tight correlation, would be in the constrained optimization procedure of Chapter 8. In principle, the optimization procedure could use either sta-

tistical estimate for its objective function. Since the *MSWL* estimate is simpler to compute, it would be preferable to the *SWL* for the objective function. In Chapter 8, this estimate will be used as the basis for finding a minimum coefficient wordlength structure.

Figure 6-3 shows a plot of the execution times of the *TWL*, *SWL*, and *MSWL*

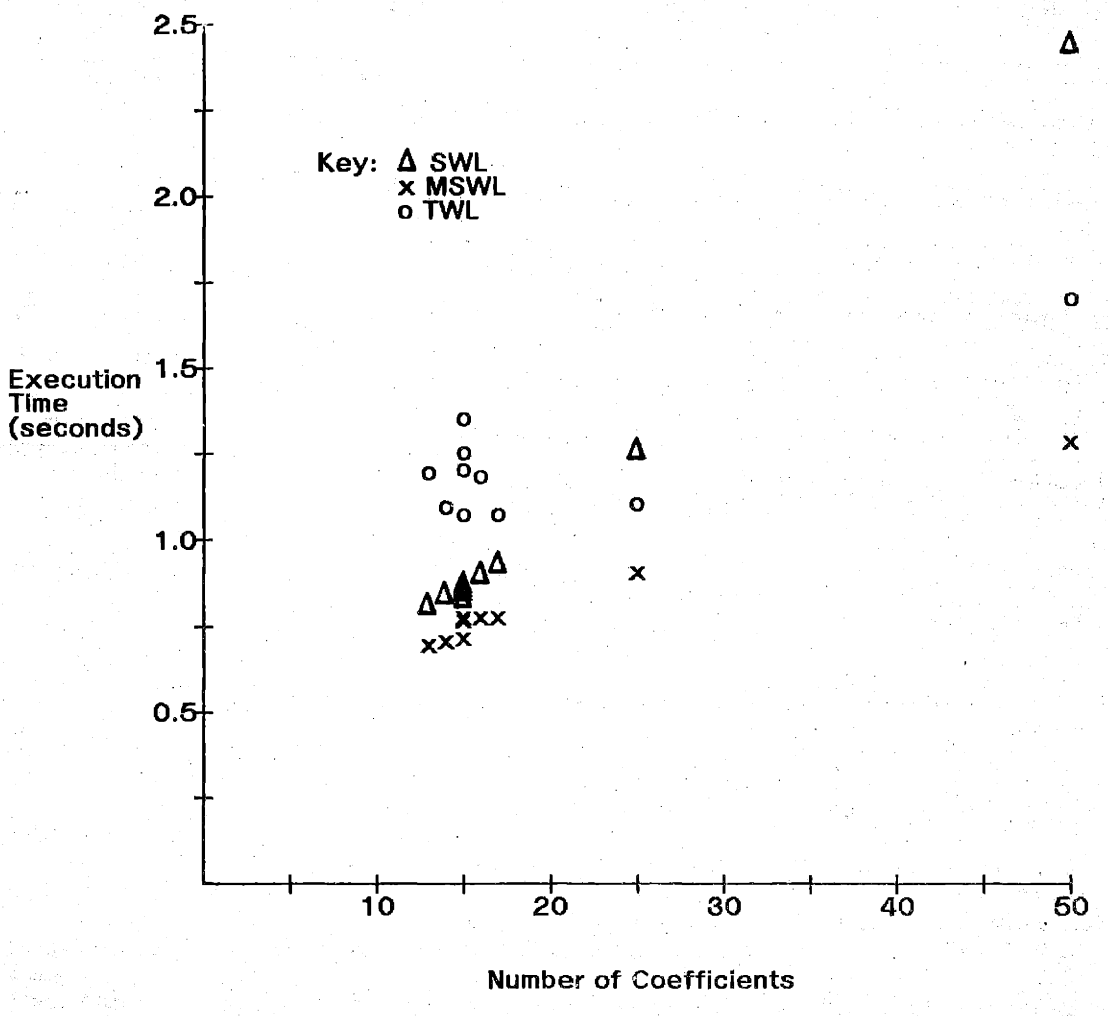



Figure 6-3: Execution times vs. Number of Coefficients

routines, as run on an Amdahl 470 at the Charles Stark Draper Laboratory, versus

the number of coefficients in the structure. From this figure we can see that the *TWL* computation takes between 1.1 and 1.35 seconds (since the routine could be written more efficiently to reduce the execution time for structure (j)). For approximately 20 coefficients or less, the *SWL* estimate is somewhat faster to compute than the *TWL* value, about 15% to 30%, and the *MSWL* is at least another tenth-second faster than this.

One important advantage to either the *SWL* or *MSWL* estimates is the second-order sensitivities they produce. Once these values are computed, it is easy to see which coefficients dominate, as far as the required coefficient wordlength is concerned. The portion of the structure in which these coefficients occur is then a likely candidate for optimization as described in Chapter 8. Specifically, the second-order section in which these coefficients occur should be *unconstrained* — in other words, it should not have a direct form II structure, but have a structure with more coefficients and thus more degrees of freedom. The optimization procedure will then exploit these extra degrees of freedom and produce an overall structure with a lower required coefficient wordlength.

In addition, there is a further advantage to knowing the individual sensitivities. In figure 6-2 we can see that structures (a), (c), (d), (e), (g), (h), and (i) have at least one large coefficient that requires the large number of integer bits (more than 1) in the fixed-point coefficient word. By replacing each of these coefficients by a smaller-magnitude coefficient followed by a shift, we can reduce the number of integer bits that are required. The amount of the shift (number of bits) will be limited by the coefficient sensitivities. For example, structure (d) has only 2 coefficients larger than two (see Appendix A). Their ideal values are approximately 15.7 and -15.7. From the *SWL* analysis, their sensitivities $\frac{\partial^2 J}{\partial c_l^2}$ are

approximately 0.043. The dominant sensitivities with respect to determining the actual coefficient wordlength (and for the sake of this discussion we will leave out the coefficients 1.99383281 and -.9938344 mentioned above) are on the order of 150. Since each factor of 2 decrease in a coefficient value results in a factor of 4 increase in its sensitivity (because we are taking second-order sensitivities), we can decrease these two large coefficients by a factor of 8 (three bits), while only increasing their sensitivities to about 2.8. Since this is still insignificant with respect to 150, the statistical (and true) fractional wordlengths will not increase appreciably. The net result is a savings in total wordlength of three bits (from 13 to 10 total bits), while adding only two simple three-bit shifts to the hardware. Note that such a shift operation does not involve any additional hardware, but just a rewiring of the respective multiplier output and the following quantizer or adder input. In structure (d), all we are doing is replacing a multiplication by 15.7173777648272 with a multiplication by 1.9646722206034 and a three-bit shift (a multiplication by 8) and similarly for the other large coefficient. The table presented in figure 6-4 shows the reduction possible for all ten structures (where this method applies). Structures (c) and (e) now rate so much better in terms of required wordlength that they are nearly as good as the best choices (b) or (f).

| structure | | I | *TWL* (no shifts) | possible shift (bits) | expected *TWL* (with shifts) |
|---|---|---|---|---|---|
| (a) | direct form II | 16 | 32 | 11 | 21 |
| (b) | parallel direct form II | 1 | 6 | unnecessary | 6 |
| (c) | parallel direct form II | 4 | 11 | 3 | 8 |
| (d) | parallel direct form II | 4 | 13 | 3 | 10 |
| (e) | parallel 1-level from (c) | 3 | 10 | 2 | 8 |
| (f) | block optimal parallel | 1 | 7 | unnecessary | 7 |
| (g) | cascade, direct form II | 11 | 21 | 6 | 15 |
| (h) | cascade, direct form II | 6 | 14 | 3 | 11 |
| (i) | cascade, direct form I | 9 | 20 | 4 | 16 |
| (j) | simple | 1 | 9 | unnecessary | 9 |

Figure 6-4: *Shifting to Reduce Coefficient Wordlength*

## §6.7 Joint Analysis of Roundoff Noise & Coefficient Rounding Effects

Chapters 5 and 6 have presented analyses of roundoff noise effects and finite coefficient wordlength effects as if the two were completely independent. Ideally, one would want to analyze the roundoff effects on a structure using its actual finite wordlength coefficients. However, the structure must of course be scaled *before* the coefficient wordlength analysis can be carried out. Thus, a near-ideal structural selection procedure would first *scale* the structure, *then* compute its required coefficient wordlength, round the infinite-precision coefficient values to that wordlength, and finally compute the necessary signal variable wordlength via a roundoff noise analysis using the rounded coefficient values. The procedure we have followed differs in that the roundoff analysis is performed using the infinite-precision coefficient values, rather than the rounded values. This simplification was made for two reasons. First, the effect of using infinite-precision coefficients in the roundoff analysis causes only very minor changes as compared to using the finite wordlength coefficients (a **second-order effect**).

Second, the nature of the roundoff analysis procedure is *approximate* to start with. We are adding just one more small approximation. Once the roundoff analysis procedure of Chapter 5 and the statistical coefficient wordlength determination methods of Chapter 6 are used to select one from a group of candidate structures, then it would be advisable to go back and do a more careful analysis of the finite wordlength effects and required wordlengths for this structure.

A more important observation is the following; we have assumed that the increase in $J$ due to roundoff noise (including the A/D contribution) must be limited to some level, say 5% of the ideal $J$, and that the increment due to finite wordlength coefficients must also be limited to some level $E_0$, say 5%. Thus the total degradation will be approximately the sum of these values, or 10%. There is no implicit reason why the overall error budget must be split evenly between these two effects. In fact, once a structure is selected using the techniques described in Chapters 5 and 6, the respective required wordlengths can be modified, perhaps to *convenient* or *more nearly equal* values by apportioning the two error limitations differently. Such a degree of freedom should be exploited to help simplify the hardware by conforming to more standard wordlengths and thus less expensive and more available hardware components.

## §6.8 Summary

In this chapter, we have examined the coefficient wordlength issue for digital feedback compensators. The use of a statistical approach to the determination of an acceptable wordlength was stressed. The common digital filtering estimate was shown to be inadequate for LQG compensators due to the optimal nature of an LQG design. Through the inclusion of second-order sensitivities in the

statistical formulation, we derived a statistical estimate that is appropriate to the LQG problem, and in fact any design problem involving the optimization of its performance criterion. As a comparison, a direct method for determining the required coefficient wordlength was presented, and 10 example structures were compared.

Based on the results presented in section 6.6, we can conclude that the *SWL* or *MSWL* estimates are not simple enough to overwhelmingly justify their use (instead of the *TWL* calculation) on a calculation-time basis *alone*. However, there are two excellent advantages for which we highly recommend their use. First, the resulting second-order sensitivities are an excellent guide for (1) reducing the required wordlength of certain structures with large coefficients (greater than two), and (2) discovering which sections of a structure dominate in determining the required wordlength (this information could be used to select which portion of a structure to optimize, as discussed in Chapter 8). Secondly, through the use of the *MSWL* as an objective function, we can effectively determine a constrained minimum coefficient wordlength structure by applying transformations as described in Chapter 8. Once a set of candidate structures has been compared with regard to their roundoff noise, coefficient wordlength effects (using the statistical estimates), precedence levels, and so forth, and a structure selected, we should analyze it in more detail. Specifically, it would then definitely be worthwhile to evaluate the *TWL* as a final step in determining the required coefficient wordlength.

# Chapter 7: Finite Wordlength Effects: Limit Cycles

## §7.1 Introduction

The roundoff noise analysis of Chapter 5 depends on the validity of the additive white noise model for roundoff quantization. However, this model is not always valid. In particular, a digital structure can exhibit oscillations known as *limit cycles*. Any linear system including one or more nonlinearities can exhibit autonomous oscillations due to those nonlinearities. For digital filters or compensators, quantization nonlinearities exist after each multiplication product or sum of products, and overflow nonlinearities exist after each adder. In addition, both nonlinearities operate on the ideal A/D converter output. We can classify the resulting oscillations as quantizer limit cycles or overflow limit cycles, depending on the type of nonlinearity that causes them. Of these two types, the overflow limit cycle tends to be more disastrous in its deleterious effect on performance — when it occurs, it has an amplitude equal to the maximum representable digital signal.

In the digital signal processing literature, there are a great number of results concerning limit cycles. An excellent review of this literature on limit cycles can be found in Kaiser [21], or in the finite wordlength survey articles by Classen, Mecklenbräuker, and Peek [60] and Oppenheim and Weinstein [57]. Willsky [16] presents a comparison of these results to the nonlinear system stability results known to the control and estimation field. Rather than cataloguing all the different results and techniques used for dealing with limit cycles in digital filters, our effort will be confined to only the more general approaches, since they are more likely to extend to the control environment.

Several points concerning the digital signal processing limit cycle results should be mentioned. First, most of these results concern zero-input limit cycles, oscillations that occur when there is no input driving the filter. When a non-zero input *is* present, it is unclear just what limit cycle behavior means, since the response of the filter to the input can be superimposed on an oscillation, or it can actually *eliminate* the oscillation [79]. Second, most of the digital filtering limit cycle results are specific to a single structure, usually the second-order direct form II structure. Since limit cycles can only be caused by nonlinearities in the recursive part of a filter, these results are further specific to the pole section of the direct form II structure. Two general conclusions follow from the digital filtering results. First, for avoiding quantizer limit cycles, sign-magnitude truncation is to be preferred over roundoff. Recall that the reverse is true when quantization noise minimization is considered. Second, for avoiding overflow limit cycles, the saturation characteristic is to be preferred over the two's complement overflow characteristic. For overflow, it is important to keep in mind that the two's complement characteristic requires no additional hardware — it is implicit in any addition using two's complement arithmetic. Additional hardware is required to implement the saturation characteristic.

As a whole, our results concerning limit cycles in digital feedback compensators are limited. However, in this chapter we will make four observations. First, we will point out that zero-input limit cycles *always* occur for control systems with open-loop unstable plants. Second, we will stress just how the feedback loop of a control system can alter the limit cycle performance of a digital compensator. In fact, even if the compensator *alone* has no limit cycles, the feedback system of plant and compensator together *can* exhibit limit cycles. Third, for a variety of

reasons, we will show that the limit cycle results in digital signal processing do not generally apply to the control setting. Finally, we will discuss the significant question of whether limit cycles themselves are an issue at all for LQG Systems. At even the simplest level, no LQG system could even be thought of as zero-input, given the system driving and measurement noises.

The remainder of this chapter is organized as follows. Sections 7.2 and 7.3 will present the more general digital signal processing approaches for dealing with quantizer limit cycles and overflow limit cycles, respectively. Finally, section 7.4 will consider the various aspects of the limit cycle issues as they concern digital feedback compensators. Specifically, the observations mentioned above will be dealt with in greater depth.

## §7.2 Quantizer Limit Cycles

There are three basic approaches for dealing with the limit cycles caused by the quantization nonlinearities in a digital structure. The first of these is simply to apply general *nonexistence* results, which guarantee that limit cycles do not occur. Many of these are so general as to apply to the overflow case as well. This procedure can be quite restrictive as to the types of structures and quantizers (roundoff or sign-magnitude truncation) that apply. The second approach is quite different; if we can bound the magnitude of the quantization effects (this bound would include limit cycle *and* noise effects) to some level dependent on the wordlength, then we need only use wordlengths long enough to make these effects negligible. Such analysis techniques are frequently based on Lyapunov theory [16]. Finally, the last procedure involves *random rounding*; basically this refers to the technique of adding randomness at selected points in a structure to

break up potential limit cycles. Of course this technique tends to add noise to the system, requiring longer wordlengths to restore performance to desired levels. All three of these methods will be reviewed in this section, and their extensions to the LQG control problem considered.

### §7.2.1 General Nonexistence Results

We will discuss three general nonexistence results described in the digital signal processing literature. The first of these is a frequency-domain criterion introduced by Claasen, Mecklenbräuker, and Peek [80] and based on the *sector* nature of the quantizer and/or overflow nonlinearities. Let us divide the digital filter under consideration into its linear and nonlinear portions as in figure 7-1. In

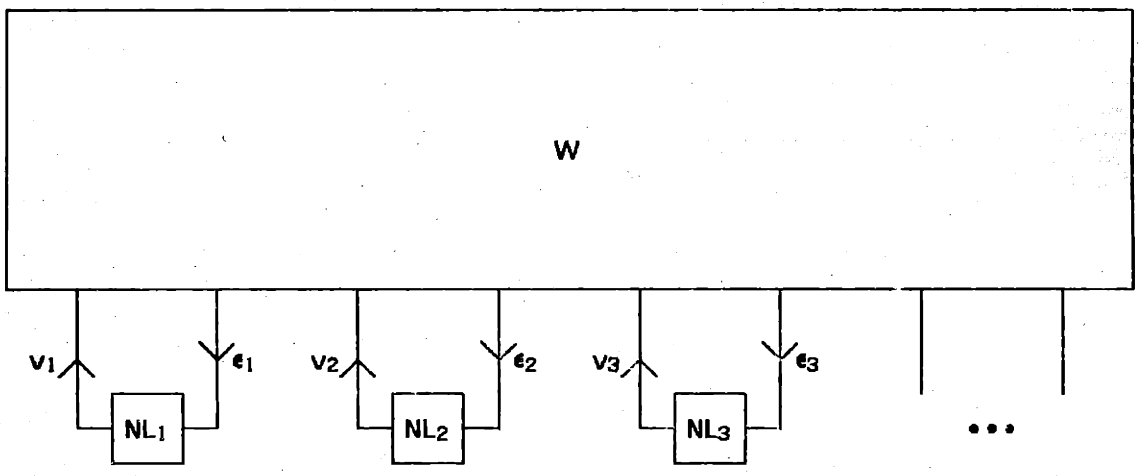



Figure 7-1: System Divided into Linear and Nonlinear Portions

general, multiple nonlinearities must be considered. The signals $e_i(k)$ and $v_i(k)$ will represent the input and output of the $i^{th}$ nonlinearity. The linear portion of the system in figure 7-1 can be described by the transfer response matrix $W(z)$,

where

$$\mathcal{E}(z) = W(z)\,V(z) \tag{7.1}$$

and $\mathcal{E}(z)$ and $V(z)$ are the $z$-transforms of $\epsilon(k)$ and $v(z)$ respectively. Now let us assume that the $i^{th}$ nonlinearity is a *sector* nonlinearity; that is, it lies entirely within the shaded sector of figure 7-2, where $m_i$ is the sector slope. (For

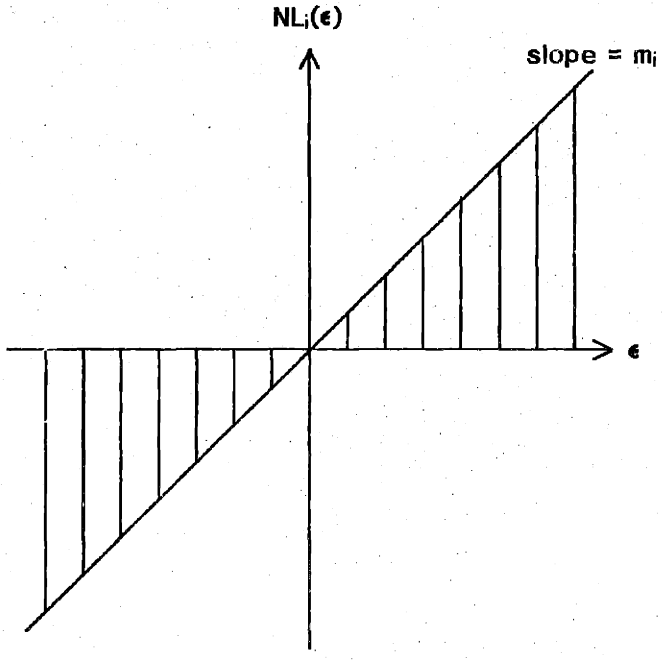



Figure 7-2: Sector Nonlinearity

roundoff quantization, $m_i = 2$, and for sign-magnitude truncation or overflow nonlinearities, $m_i = 1$.) The result derived in [80] states the following: given $k_0$ nonlinearities as described above, and a $W(z)$ that is finite for $|z| = 1$, zero-input limit cycles of period $N$ are absent if:

$$\text{Re}\left\{W\left(e^{j2\pi h/N}\right) - \text{diagonal}\left(\frac{1}{m_i}\right)\right\} < 0 \qquad (7.2)$$

$$\text{for } h = 0, 1, \cdots \text{ integer}\,[N/2]$$

Furthermore, if the nonlinearities are also time-invariant, with a symmetric nondecreasing characteristic, then limit cycles of period $N$ are absent if the real part of

$$\left\{I_{k_0} + \text{diag}\left[\sum_{j=1}^{N-1}\left(\alpha_{ji}\left(1-z_h^j\right) + \beta_{ji}\left(1+z_h^j\right)\right)\right]\right\} W(z_h) - \text{diag}\left(\frac{1}{m_i}\right) \qquad (7.3)$$

is negative definite ( $< 0$), for all $\alpha_{ji}$ and $\beta_{ji}$ greater than or equal to zero and

$$z_h = e^{j2\pi h/N}$$

Equation (7.3) is more difficult to apply than (7.2) since linear programming techniques must be used to take advantage of the $\alpha$ and $\beta$ parameters. However, (7.3) is a more useful condition, since it may prove nonexistence when (7.2) does not. (Note that for $\alpha = \beta = 0$, conditions (7.2) and (7.3) are identical.) Unfortunately, both these relations require multiple evaluations (one per $N$), not to mention the task of proving negative definiteness. We can simplify the application of (7.2) and (7.3) somewhat by expressing these conditions differently. Šiljak [81] has found an efficient technique for proving the positive realness of a function $G(z)$, which he has extended to the matrix $G(z)$ case. (A real rational function $G(z)$ is strictly circle positive real if it has no poles outside the unit circle, and the real part of $G(z)$ is strictly positive on and outside the unit circle.) Thus, for one nonlinearity, we could replace the repeated evaluation of (7.2) with a test for

the positive realness of $\left\{\frac{1}{m_l} - W(z)\right\}$. Still, this procedure is not terribly simple, especially in the matrix case. Application of (7.2) and (7.3) to the 1 or 2 quantizer two-pole direct form II sections of figure 7-3, for both roundoff and sign-

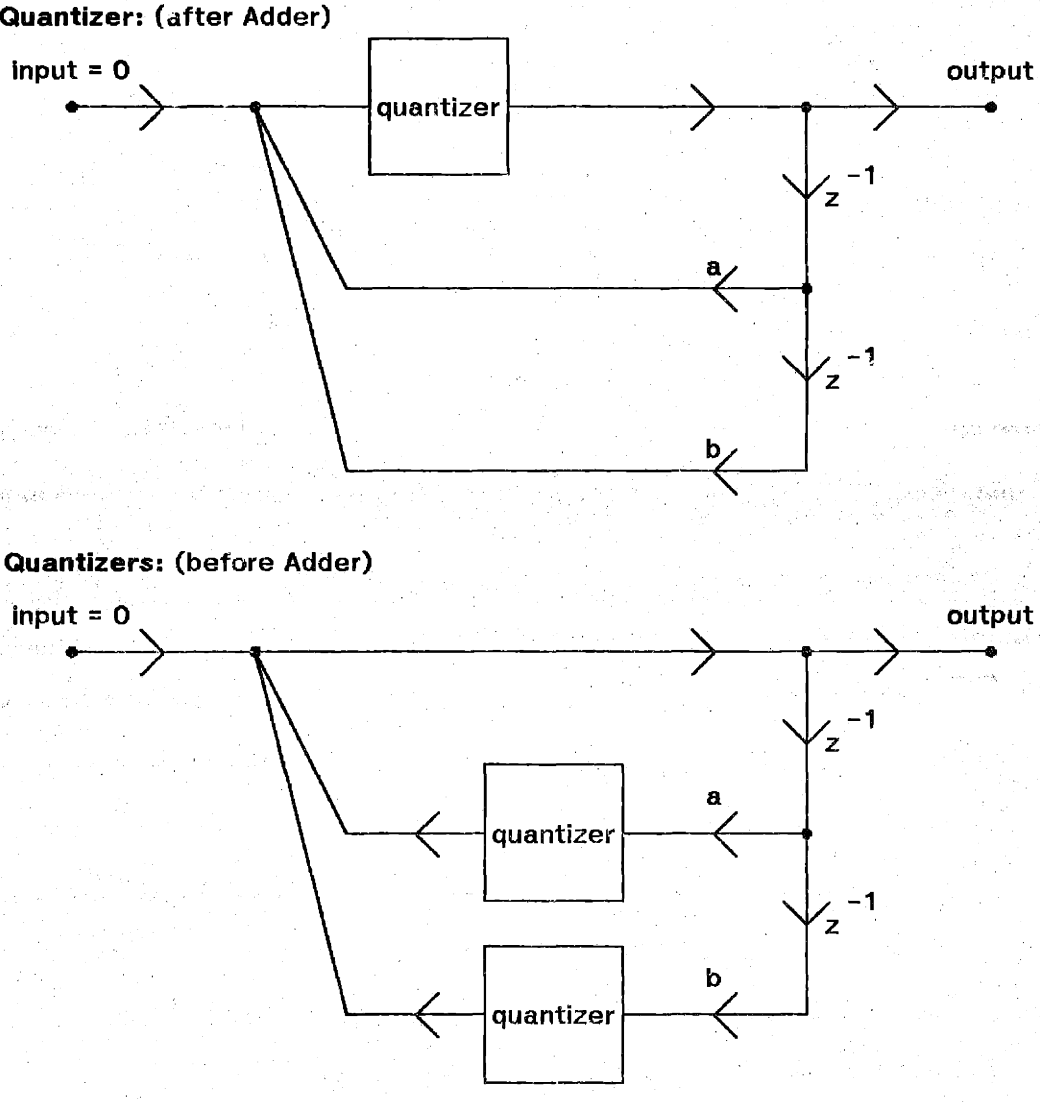



Figure 7-3: Direct Form-II (no zeros); 1 and 2 Quantizers

magnitude truncation nonlinearities, shows the advantage in using sign-magnitude

truncation over roundoff; the range of possible $a$ and $b$ values for which a quantizer limit cycle cannot occur is much greater under sign-magnitude truncation quantization [21,60].

A different limit cycle nonexistence result for digital filters can be related to the norm of the transition matrix of one-level state space structures [82,83,84,85,86]. This procedure can be applied to either the quantizer or overflow limit cycle. Suppose we have a one-level state space digital filter structure:

$$\begin{aligned} v(k+1) &= f\Big\{ Av(k) \Big\} + B\, y(k) \\ u(k) &= C\, v(k) \end{aligned} \tag{7.4}$$

where $f$ represents all the nonlinear operations of the compensator. Note that the type of nonlinearity implied by (7.4) can act *only* on the ideal values $A\, v(k)$. Thus quantization must occur *after* addition, implying double-precision adders, and similarly for the overflow nonlinearity. For two's complement overflow (see section 7.3), this requirement presents no difficulty; if we define $Q(\,)$ to represent the two's complement nonlinearity, the following relationship is true: [82]

$$Q(\eta_1 + \eta_2 + \eta_3) = Q(\eta_1 + Q(\eta_2 + \eta_3)) \tag{7.5}$$

where $\eta_i$ is the result of a multiplication. The same cannot be said of the saturation overflow characteristic, and one or two extra adder bits are required to accumulate the true sum before applying saturation.

Let us consider the zero-input case ($y(k) = 0$) for (7.4). For quantizer and overflow nonlinearities, we can show that:

$$\|f(v)\|_2 \le \gamma \|v\|_2 \quad \text{for all } v \tag{7.6}$$

where $\|v\|_2$ refers to the Euclidean norm $(v' v)^{1/2}$. For sign-magnitude truncation and all common overflow characteristics, $\gamma$ would be 1, while for roundoff $\gamma$ would be 2.

If we define the matrix norm of $A$ as follows: [83]

$$\|A\|_2 = \max_{x \neq 0} \left\{ \frac{\|A v\|_2}{\|v\|_2} \right\} \tag{7.7}$$

then we can write: [83]

$$\|A v\|_2 \le \|A\|_2 \|v\|_2 \tag{7.8}$$

Combining (7.4), (7.6), and (7.8) produces:

$$\|v(k+1)\|_2 \le \gamma \|A\|_2 \|v\|_2 \tag{7.9}$$

Thus we can ensure the nonexistence of zero-input limit cycles by the condition

$$\gamma \|A\|_2 \le 1 \tag{7.10}$$

since this implies a continuously-decreasing state norm. Mills, Mullis, and Roberts [82] have expressed this result in a different manner for the more general case of $\|v\| = (v' D v)^{1/2}$ where $D$ is a positive definite diagonal matrix, and the case of an overflow nonlinearity ($\gamma = 1$): overflow (and hence sign-magnitude truncation with double-precision adders) limit cycles will not occur if and only if $D - A' D A$ is positive-definite. (This result is based in Lyapunov theory.)

Based on these results, it is natural to consider structures for which the norm of $A$ is small (and of course less than 1). It can be shown that a minimum

norm filter would be one for which:

$$\|A\|_2 = \max_i \left\{ |\lambda_i| \right\} \tag{7.11}$$

This quantity is always less than 1 for (stable) digital filters; thus such filter structures have no overflow oscillations, and no quantizer oscillations under sign-magnitude truncation.

Barnes [84] discusses minimum norm filters composed from minimum norm sections of arbitrary order. However, we will restrict our attention to the more useful case of second-order sections. In fact, a minimum norm second-order section is identical to the Rader and Gold coupled form section mentioned in Chapter 6. The matrix $A$ for a coupled form section with poles at $(\sigma \pm j\omega)$ would appear as follows: [85] (See figure 7-4.)

$$A = \begin{bmatrix} \sigma & \omega \\ -\omega & \sigma \end{bmatrix} \tag{7.12}$$

The lack of limit cycles under overflow and sign-magnitude truncation for the this structure will not be affected by scaling [82].

For roundoff quantization, these norm-based results cannot be used to prove the nonexistence of limit cycles for the minimum norm structure unless the maximum filter eigenvalue is less than one half. In fact, Jackson has shown that roundoff limit cycles *will* occur for the coupled-form structure [86]. Fam and Barnes [85] have introduced a method for taking a filter structure whose $A$ norm is greater than one half, and computing an equivalent form whose norm is less than one half. This technique combines recursive and nonrecursive filter sections but greatly increases the number of multipliers and delays over the original struc-

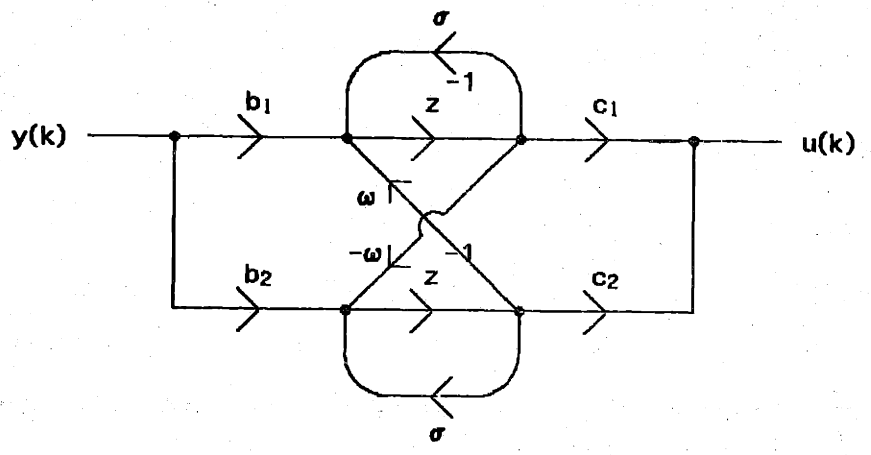



Figure 7-4: Coupled Form (Normal) Second-Order Section

ture.

It should be mentioned here that these results tie directly into the results for wave digital filters. Fettweis and Meerkötter [47] have shown through the use of a state-norm called *pseudopower* that overflow limit cycles and quantizer limit cycles will not occur in wave digital filters using sign-magnitude truncation quantization and any common overflow characteristic, such as two's complement or saturation.

### §7.2.2 Limit Cycle Amplitude Bounds

One common method for dealing with quantizer limit cycles is to bound their amplitude, and then to choose a wordlength long enough to make this bound small. Many methods exist for formulating amplitude bounds on the effects of quantization, which of course must include limit cycle effects. A good review of these methods, many of which have been presented in the context of sampled-data control systems, can be found in [60] and [87]. In the results pertaining to digital

filters [87,88,89], the direct form II second-order section is usually considered, or specifically the recursive portion of this section. Recall that only the nonlinearities in the recursive portion of a structure can give rise to limit cycles. Of course this simplification is not possible for a control system, since the entire compensator structure is involved in the feedback loop.

We will discuss one of the more general approaches to limit cycle amplitude bounding. This approach involves the use of Lyapunov theory, and is considered for digital filters in [87] and for sampled-data control systems in [11] and [12]. Consider a system with the following state equation:

$$x(k+1) = A\,x(k) + B\,u(k) \tag{7.13}$$

where $x$ represents the state, and $u$ the inputs. Following Parker and Hess [87], the system (7.13) is bounded-input bounded-output stable if the (zero-input) system

$$x(k+1) = A\,x(k) \tag{7.14}$$

is asymptotically stable in the large. If so, a Lyapunov function $x'\,P\,x$ exists where $P$ is the symmetric positive-definite solution to the equation:

$$P = A'\,P\,A + C \tag{7.15}$$

for any symmetric positive-definite matrix C. If the input to the system (7.15) is upper bounded by some constant $\kappa$, then an upper bound on the norm of the state vector $x$ can be derived [11,12]. This bound, which again will include *all* the effects of quantization, is fairly complex to compute, is a function of $A$, $B$, $P$, and the eigenvalues of the C and P matrices, and will be directly proportional to $\kappa$.

The procedure outlined above can be easily applied to digital filters [87] with one or more precedence levels. For the roundoff nonlinearity, we know that

every roundoff quantization error is bounded by $\frac{\Delta}{2}$ ($\Delta$ for sign-magnitude truncation). We can simply define these quantizer errors as inputs to the filter system, and then compute an upper bound on the filter state norm that is proportional to $\Delta$. The difficulty that arises in using this bound is in selecting a Lyapunov function, or equivalently, in selecting $C$ [87]. Consequently, this bound can be quite loose, especially for certain combinations of filter parameters [87].

Other methods of computing limit cycle amplitude bounds either are even less tight then the Lyapunov-based bound ([9,10]), or are not easily extendible to the control system setting (such as the effective value method of Jackson [88]), or are even more difficult to compute (such as the matrix method of Parker and Hess [87]).

### §7.2.3 Random-Rounding Techniques For Limit Cycle Quenching

The previous two sections have described two different ways for dealing with limit cycles. The first involved using structures for which limit cycles could be proven not to exist. The second involved the use of sufficient signal bits to bound the limit cycle amplitude to a negligible level. A third method exists - eliminating limit cycles when they do occur (presumably determined by simulating the structure). The idea behind this procedure is that limit cycles (which represent a correlated quantizer error effect), can be *broken up*, or *decorrelated* by introducing some randomness into the quantization procedure. This procedure results in the replacement of a periodic limit cycle by an aperiodic sequence of reduced power [90]. Justification for this method can be found in Kieburtz [79], who reported limit cycle breakup as the level of a random input signal was raised. Further intuition for the technique can be presumed from the success enjoyed by *dith-*

*er* techniques for the stabilization of unstable nonlinear systems. [91,92]

Specific results concerning the use of randomized quantization methods exist only for the case of the direct form II second-order section. The first method involves randomly switching between roundoff quantization and sign-magnitude truncation. By utilizing roundoff *most* of the time, its low-noise advantages can still be maintained, while the occasional use of sign-magnitude truncation will give us the reduced number of limit cycles common to this type of quantizer. Kieburtz, Lawrence and Mina [90] outline this method and present specific examples of its use. Unfortunately, such a technique cannot be guaranteed to eliminate all DC and half-rate limit cycles (limit cycles with a two-sample period). However, Lawrence and Mina [93] do describe some additional constraints that can be added to prevent such limit cycles.

Büttner [94] has taken a different approach to implementing random quantization. In his approach, a random signal is *injected* at one point in the direct form II structure to break up any possible limit cycle. One obvious difference with this approach is that, with no input to the filter, there will still be a noise output. In a control system, already driven by noise, this additional noise would probably be insignificant. Specifically, Büttner describes two possible approaches; first, in the direct form II section with only one quantizer (after a double-precision addition), simply replace the least significant bit of the quantized sum with a *random* bit. This procedure produces 4 times the output noise power as compared to rounding, since the error introduced can be anywhere between $\pm\Delta$, but has the advantage of eliminating all possible limit cycles. The second approach introduces a random least significant bit in one of the products input to the double-precision adder. Although this generates approximately half the noise generated by the first

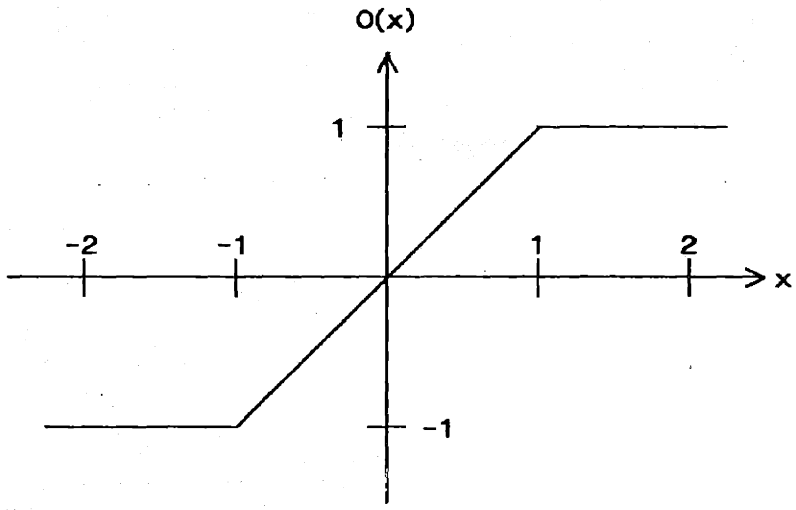
method, it will not prevent the occurrence of limit cycles unless the input to the second-order section is aperiodic and non-constant. Büttner then recommends using a cascade of second-order sections, with the first approach used to suppress all limit cycles in the first section, and the second *lower-noise* approach used in all remaining sections. Since the input to these sections must contain the random output component generated by the first section, the second method will be sufficient to suppress all limit cycles in these sections. Examples were presented comparing this random rounding approach to the use of sign-magnitude truncation to eliminate limit cycles, and also to the use of roundoff quantization with longer wordlengths to reduce limit cycle amplitude. Again, all these results were generated only for structures composed of direct form II second-order sections.

## §7.3 Overflow Limit Cycles

In this section, we will examine the results specific to overflow limit cycles. Overflow limit cycles are particularly important because they have maximal amplitude — thus, of course, bounding techniques do not apply. In general, there are two overflow characteristics of particular interest, saturation (figure 7-5a) and two's complement (figure 7-5b). A two's complement overflow characteristic is the natural overflow characteristic resulting when using two's complement addition. No additional hardware is necessary to realize this overflow nonlinearity. The saturation overflow nonlinearity, which does require some hardware, is less prone to causing overflow limit cycles than the two's complement characteristic.

Two separate issues concerning overflow have been discussed in the digital signal processing literature, the prevention of zero-input overflow limit cycles, and forced-response stability. Stability of the forced response means that the

(a) Saturation:




(b) Two's Complement:

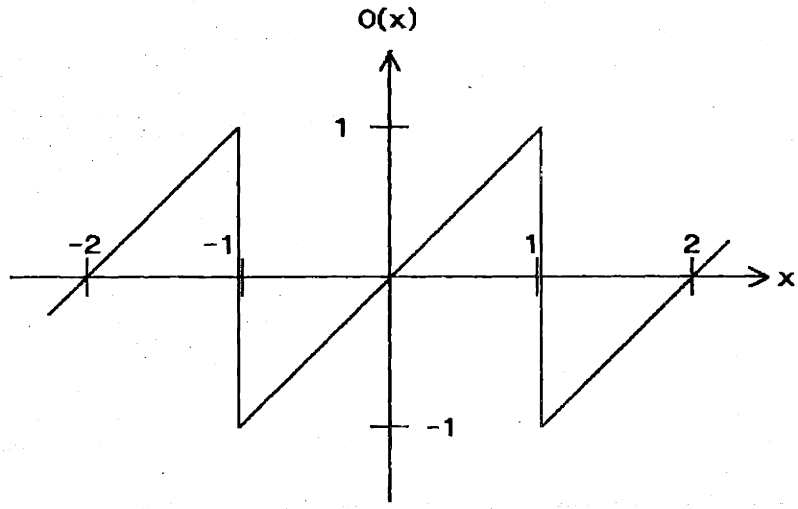



Figure 7-5: Common Overflow Characteristics

filter must *recover* from an overflow, that is, return asymptotically to the state values that would have occured if no overflow nonlinearity had been present.

General results concerning zero-input overflow limit cycles can be inferred

from the discussion in section 7.2.1 on the frequency-domain criterion of Claasen, Mecklenbräuker, and Peek for the saturation nonlinearity (using $m_i = 1$). Using the norm-based method of Barnes and Fam, or Mullis and Roberts, we can generate nonexistence results that would apply to *any* common overflow characteristic.

More specific results exist for the second-order direct form II section

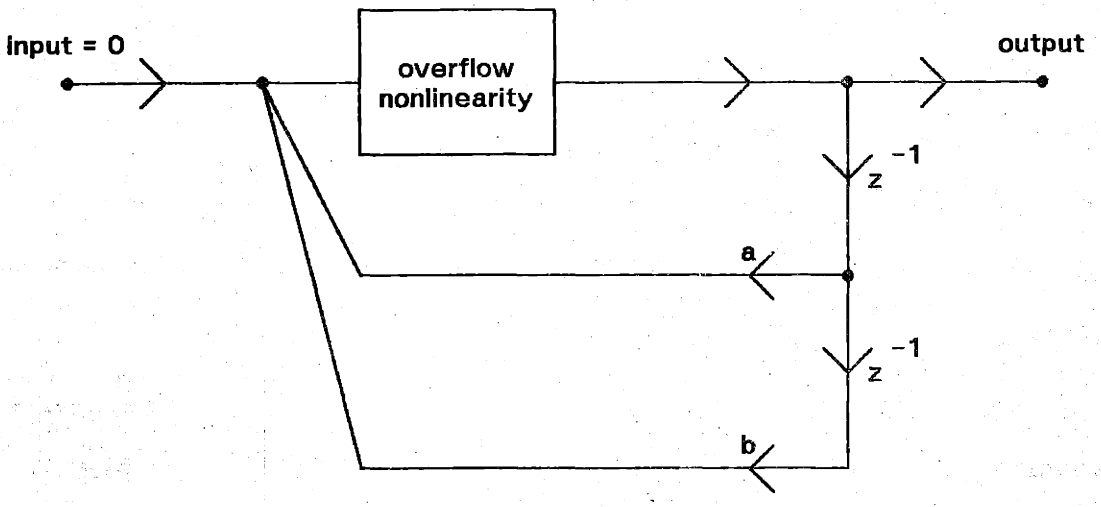



Figure 7-6: Direct Form II with Overflow Nonlinearity

shown in figure 7-6 and for structures composed of such sections. Willson [95] and Ebert, Mazo, and Taylor [96] have found regions in the *a*, *b* parameter plane where overflow limit cycles will not occur with two's complement overflow, and have shown that *no* limit cycle can occur when using the saturation overflow characteristic for any (stable) values *a*, *b*. In general the saturation characteristic is to be preferred over the two's complement characteristic so far as overflow limit cycles are concerned. However, it does require extra hardware components to implement the saturation overflow characteristic. Thus we would test the general conditions in section 7.2.1 to see whether or not the use of two's comple-

ment overflow could cause oscillations. The use of the saturation overflow characteristic, with its additional hardware, would be advised whenever the general criteria of section 7.2.1 did *not* succeed in guaranteeing the absence of limit cycles for the two's complement characteristic.

Recovery from overflow can be determined by the following general result also derived by Claasen, Mecklenbräuker, and Peek [97]: if a system has no zero-input overflow limit cycles for all time-varying nonlinearities satisfying:

$$-m_j \le \frac{O(x,k)}{x} \le 1 \quad \text{for } x \neq 0, \text{ and } m_j > 0 \text{ for all } k \tag{7.16}$$

where $O(\,)$ is the overflow nonlinearity, and this condition could possibly be tested using the general criteria described in section 7.2.1, then the forced response will be stable for all overflow nonlinearities satisfying (see the shaded portion of figure 7-7):

$$\begin{aligned} 1 + m_j - m_j\, x < O(x) \le 1 \quad & \text{for } x > 1 \\ -1 - m_j - m_j\, x > O(x) \ge -1 \quad & \text{for } x < -1 \end{aligned} \tag{7.17}$$

This result means that a system with no zero-input overflow limit cycles for *all* overflow characteristics satisfying (7.16) for $m_j$=1 (such as the wave digital filter) will be forced-response stable for characteristics satisfying (7.17). Saturation satisfies (7.17), but two's complement overflow does not. Again, this result demonstrates the general advantage of saturation over two's complement overflow so far as limit cycles are concerned.

Beyond the general result of (7.16) and (7.17), there also exist specific results concerning forced-response stability for the direct form II second-order section of figure 7-6 [98,99].

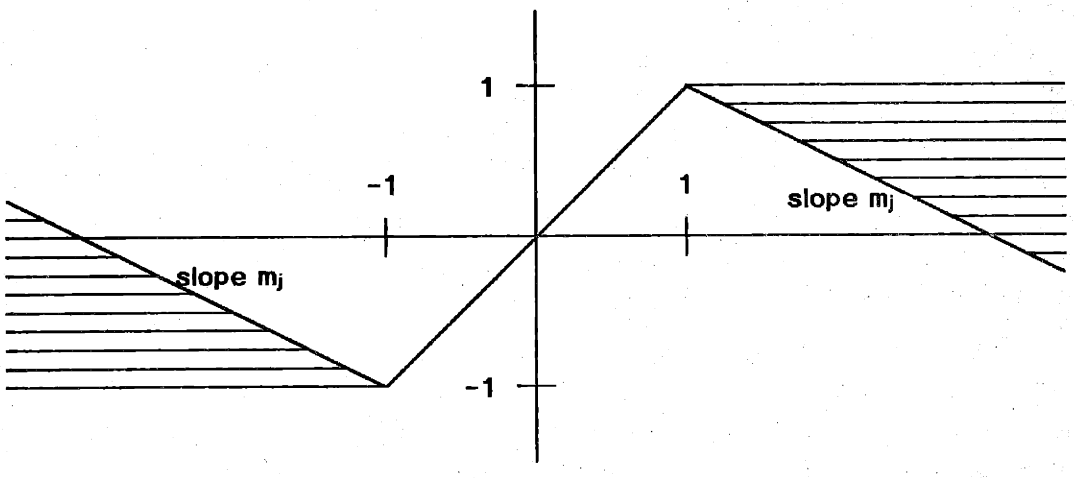



Figure 7-7: Forced-Response Stable Overflow Characteristic

## §7.4 Digital Feedback Compensator Limit Cycles

In this section we will consider the limit cycle issue as it relates to digital feedback compensators. Several important observations can be made. First, any digital control system with an open-loop unstable plant *must* exhibit quantizer limit cycles. Recall that the plant output is sampled, digitized, and quantized at the compensator input. This means that any output magnitude below the smallest quantization level is effectively ignored by the compensator. If the open-loop plant is unstable, the output will tend to increase in magnitude until it reaches the lowest quantization level, and some control action can occur to drive it back towards zero. However the process will then repeat. The net result is a form of low-amplitude limit cycle in the output of the system. Such a limit cycle will occur no matter what the transfer functions of the plant and compensator are, as long as a right-half plane pole exists, although these parameters will certainly affect the amplitude and frequency of the limit cycle. A proper choice of A/D wordlength

would keep this amplitude at the system noise level, so that it could essentially be ignored. One other implication of the presence of this limit cycle is that *no* general digital filtering limit cycle nonexistence result can succeed in proving limit cycle nonexistence for digital control systems with unstable open-loop plants. Furthermore, even systems with open-loop plants that have poles at $s=0$ can exhibit low-amplitude limit cycles if *any* DC offset exists in the output of the D/A converter.

One of the key points relating to compensator limit cycles is the overall effect of the closed loop on the limit cycle behavior of the compensator. For example, consider the digital compensator as a stand-alone digital network. Any limit cycles that this open-loop compensator may exhibit are strictly dependent on the nonlinearities in the recursive sections of the compensator. However, when the compensator is embedded in the feedback loop, *all* the nonlinearities are part of a recursive portion of the control system, and thus are all involved in determining limit cycle behavior. Thus, compensator limit cycles that would occur for the open-loop situation will be altered when the loop is closed. By the same reasoning, even if the open-loop compensator would not exhibit limit cycles, the overall feedback system of plant and compensator together *may* exhibit limit cycles. As an example, consider the simple control system in figure 7-8. Any finite impulse response open-loop compensator or filter is non-recursive. Therefore it can have no limit cycles. However, when we embed such a filter in a closed-loop stable control system as in figure 7-8, limit cycles may occur. For the example above, let us measure signal amplitude in units of $\Delta$, the quantization step size defined in Chapter 5. With either roundoff or sign-magnitude truncation quantization, the output $y$ can exhibit the following half-rate limit cycle:

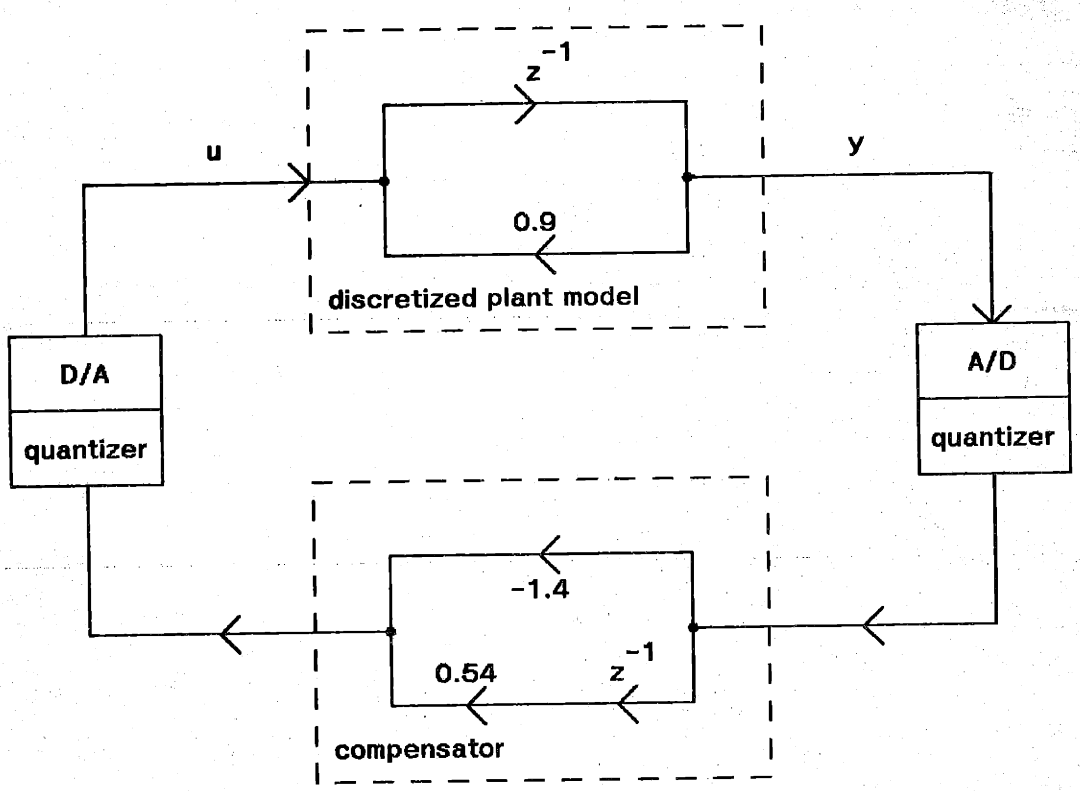



Figure 7-8: Control System with Finite Impulse Response Compensator

+10, -10, +10, -10, ...

A related limit cycle result specific to feedback systems has been reported by Fettweis and Meerkötter [100]. Motivated by the presence of digital filters in looped telegraph systems, they have shown the following. For a finite impulse response or wave digital filter embedded in a feedback loop, no quantizer limit cycles can occur if sign-magnitude truncation is used for all quantization operations including the A/D *and*:

$$\max_{|z|=1} |H_1(z)| \quad \max_{\omega} |H_2(j\omega)| < 1 \tag{7.18}$$

where $H_1(z)$ is the transfer function of the digital network embedded in the loop,

and $H_2(j\omega)$ is the transfer function of the open-loop plant. This result is quite similar to the small loop-gain theorem known to control theorists [60]. As with the digital filtering results, the above condition points out the advantage of sign-magnitude truncation over roundoff quantization so far as limit cycles are concerned. Unfortunately, for *control* systems in general, the condition (7.18) is *very* restrictive in terms of the types of plants one could consider. Certainly any system whose plant had an integrator pole or even a strong resonance would not satisfy (7.18). However, this is the only real result in the literature for quantizer limit cycles in feedback systems.

Another important observation is that the techniques for dealing with limit cycles in digital filters do not tend to work for control compensators. As shown above, *none* of the nonexistence techniques can be extended to consider open-loop unstable plants. Now let us consider control systems whose plants have integrator poles. As a simple example, consider a double-integrator plant:

$$\dot{x}[t] = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x[t] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[t]$$
$$y[t] = \begin{bmatrix} 1 & 0 \end{bmatrix} x[t] \qquad (7.19)$$

If we discretize this system at a sampling rate of 1 Hertz, and design a first-order compensator, the three-quantizer configuration of figure 7-9 results. To apply the results of Claasen, Mecklenbräuker, and Peek discussed in section 7.2.1, we must first compute the matrix $W(z)$. Defining $e$ and $v$ as shown in figure 7-9, $W(z)$ will be:

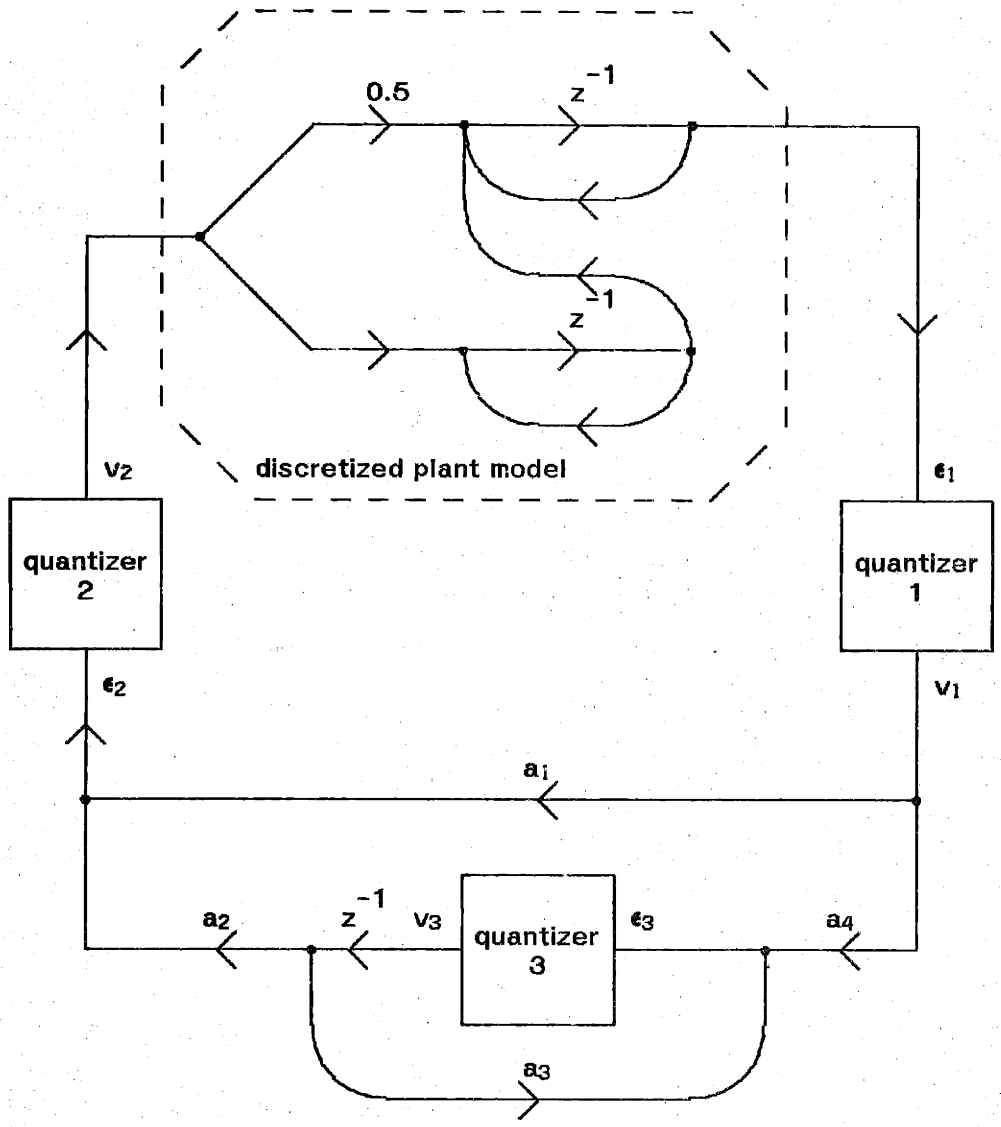



Figure 7-9: Double-Integrator Control System

$$W(z) = \begin{bmatrix} 0 & a_1 & a_4 \\ \left\{ \dfrac{z^{-1}(.5+z^{-1})}{(1-z^{-1})^2} \right\} & 0 & 0 \\ 0 & \left\{a_2 z^{-1}\right\} & \left\{a_3 z^{-1}\right\} \end{bmatrix} \qquad (7.20)$$

Unfortunately, the (2,1) entry of $W(z)$ is not finite on the entire unit circle, and thus the results of (7.2) and (7.3) cannot be applied. This will be true for any system whose plant has an integrator pole. One possible method for handling this problem would be to replace the $z=1$ poles in the $W(z)$ matrix with poles at $z=1-\epsilon$ where $\epsilon>0$. Then we could evaluate (7.2) or (7.3) in the limit as $\epsilon \to 0$. However this evaluation, or the application of the positive real test of Šiljak, will be even more complex to compute. Note that *if* a discretized plant has all its poles entirely within the unit circle, then the Claasen, Mecklenbräuker, and Peek results may be used directly.

Now let us attempt to apply the general norm-based results of section 7.2.1. To account for the behavior of the entire closed-loop system the vector $v(k)$ in (7.4) would have to include both the plant *and* compensator states. Following the analysis of (7.6) through (7.10), this would involve the evaluation of the norm of the closed-loop system matrix analogous to the matrix $A$ in (7.4), and the assumption that the nonlinearity $f$ operates on the entire vector $Av$. For the compensator case this would be a very restrictive assumption, since in fact the nonlinearity only operates on the compensator states. Furthermore, the norm-based analysis applies only to one-level structures. Also, the main advantage to the norm-based technique, namely the derivation of minimum norm structures, cannot be applied to compensator structures; it would involve transforming the

closed-loop system matrix $A$. However, this matrix is highly constrained given the control system configuration of plant and compensator. Thus $A$ can not be subject to arbitrary transformations.

The Lyapunov-based bound discussed in section 7.2.2 has been actually used for control applications [11,12], and could even be used for open-loop unstable plants. In the analysis of section 7.2.2, let us consider the performance of the entire closed-loop system. The vector $x$ in (7.13) and (7.14) would have to be replaced by a vector including all the plant *and* compensator states as mentioned above. Of course, in the LQG case, we are not interested in bounding the norm of $x$, but the more general performance index-related norm $\|x'\mathfrak{T}x\|$, where $\mathfrak{T}$ is defined in (5.34). However, since $\mathfrak{T}$ is a symmetric positive-definite matrix, it can be factored into the product $\mathfrak{T} = T'T$. Thus we can define a new $x$ to be $Tx$, similarity transform $A$ and $B$, and proceed as outlined in (7.13)-(7.15). The resulting bound will be just as loose as for the filtering case; the difficulty will still be in selecting the Lyapunov function.

The final point we would like to make concerns the general question of limit cycles in control systems. No LQG control system is actually zero-input in nature; there is always system noise present. According to the results of Büttner discussed in section 7.2.3, it is likely that this noise will quench autonomous oscillations if the noise level is large enough. Thus limit cycle oscillations themselves may not be an issue in most control systems. However, there are other effects caused by the nonlinear quantization operations in a compensator. First, jump discontinuities may occur. In such a case, small changes in the input signal lead to large jumps in the output [16]. Furthermore, we have not even considered the effects of the correlated noise that results from the presence of quantization non-

linearities. Even if limit cycles do not occur, the presence of correlated noise in control systems can significantly deteriorate performance. Recall that LQG systems are designed with the assumption that the system noises are white. This whole area is largely unexplored for digital control systems.

# Chapter 8: The Optimization of Structures

## §8.1 Introduction

Techniques for the optimization of structures with respect to some scalar objective function are very important for the synthesis of compensator structures. Typically this objective function would involve either the increase in the performance index due to roundoff noise, or some measure of coefficient sensitivity such as the *SWL* or *MSWL*, or perhaps a weighted combination of the two. In such a technique, it is important to have control over the number of multipliers and delay elements in the optimized structure, since these parameters are critical in determining the complexity of the hardware.

As shown in Chapter 3, any structure can be transformed to a new (infinite-precision equivalent) structure through the use of a set of transformation matrices. In the context of the modified state space appropriate to controllers, if we have some scaled structure with parameters $\Psi_1, \Psi_2, \cdots \Psi_q$, then we can transform this structure to one with parameters $\hat{\Psi}_1, \hat{\Psi}_2, \cdots \hat{\Psi}_q$ by:

$$\hat{\Psi}_i = P_i \Psi_i \left(P_{i-1}\right)^{-1} \quad \text{for } i = 1, \cdots, q \tag{8.1}$$

where the $P_i$ for $i = 1, \cdots, q-1$ are general non-singular transformation matrices, and

$$P_0 = \begin{bmatrix} P & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad P_q^* = \begin{bmatrix} P & 0 \\ 0 & 1 \end{bmatrix}$$

The presence of unity entries in the matrices $P_0$ and $P_q$ are necessary so that

the actual input and output nodes of the original structure are not altered by the transformation process. One consequence of this restriction is that the output node scaling parameter $\rho$ described in section 5.2 will be invariant to such transformations.

Once we have computed (8.1), the new structure will have to be *rescaled* so that it satisfies the same dynamic range constraints as the original un-transformed structure. This overall technique will result in a new structure with the *same* number of delay elements as the original. However, if the matrices $P_i$ are completely general, the number of coefficient multiplies (non-unity and non-zero entries in the matrices $\Psi_i$) will be very large. Thus it is necessary to constrain the $P_i$ matrices in order to gain control over the resulting number of coefficient multiplies.

Chan [17,101] has presented such a constrained optimization technique for digital filters, using a notation appropriate for describing digital filter structures. In section 8.2 we will present the steps involved in this constrained optimization technique for a general objective function, but in the context of the modified state space representation appropriate to digital feedback compensators (see Chapter 3). In section 8.3 we will adapt the technique of Chan for the minimization of roundoff noise effects in compensators, and apply the technique to a specific example. In section 8.4, we will use the *MSWL* estimate presented in Chapter 6 to adapt Chan's general technique to the minimization of coefficient rounding effects in compensator. No specific example will be presented. Finally, in section 8.5 we will discuss methods for selecting which entries in the original $\Psi_i$ matrices are to be constrained (held constant), and which are to be varied,

presumably becoming non-zero and non-unity. This last section represents an important extension to the work of Chan, since it applies equally well to digital compensators and to digital filters.

## §8.2 The General Constrained Optimization Technique of Chan

The optimization technique of Chan is based on the following observation [17,101] (here considered in the context of the modified state space representation). Consider the differential equation (8.2):

$$\frac{d\Psi_i(t)}{dt} = G_i(t)\,\Psi_i(t) - \Psi_i(t)\,G_{i-1}(t) \quad \text{for } 1 \le i \le q \tag{8.2}$$

where the matrices $G_i$ are of appropriate dimension. Any solution $\{\Psi_1(t), \cdots, \Psi_q(t)\}$ at any $t$ will represent a structure (infinite-precision) equivalent to $\{\Psi_1(0), \cdots, \Psi_q(0)\}$ if:

$$G_0(t) = \begin{bmatrix} G(t) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad G_q(t) = \begin{bmatrix} G(t) & 0 \\ 0 & 1 \end{bmatrix}$$

where $G(t)$ is arbitrary. The solution to (8.2) has the form:

$$\Psi_i(t) = P_i(t)\,\Psi_i(0)\,\left(P_{i-1}\right)^{-1}(t) \tag{8.3}$$

where

$$\frac{dP_i(t)}{dt} = G_i(t)\,P_i(t) \qquad \text{for } 0 \le i \le q \tag{8.4}$$

and the initial condition $P_i(0)$ matrices are identities. Starting with an initial structure which we will assume to be scaled, the technique basically integrates

(8.4) to obtain new transformed structures. The $G_i$ matrices are selected to cause an overall reduction in some objective function. Constraining any particular coefficient in a $\Psi_i$ matrix to be constant can be easily accomplished by holding its derivative in (8.2) to zero, which implies constraints on $G_i$ and $P_i$.

Now let us present this procedure in detail. Define $\nu$ to be the operation that forms a vector from a matrix by stacking its columns:

$$\nu(\Psi_i) = \begin{bmatrix} \textit{column 1} \\ \textit{column 2} \\ \vdots \\ \textit{last column} \end{bmatrix} \tag{8.5}$$

Using this operator, let us define $\psi(t)$ and $g(t)$ to be vectors composed of all the elements of $\{\Psi_1(t), \cdots, \Psi_q(t)\}$ and $\{G(t), G_1(t), \cdots, G_q(t)\}$:

$$\psi(t) = \begin{bmatrix} \nu(\Psi_1(t)) \\ \vdots \\ \nu(\Psi_q(t)) \end{bmatrix} \qquad g(t) = \begin{bmatrix} \nu(G_1(t)) \\ \vdots \\ \nu(G_q(t)) \end{bmatrix} \tag{8.6}$$

We can now express $\frac{d\psi(t)}{dt}$ as a linear function of $\psi(t)$ and $g(t)$ using (8.2) and (8.6):

$$\frac{d\psi(t)}{dt} = F(t)\, g(t) \tag{8.7}$$

where the large matrix $F(t)$ is a function of the elements of $\psi(t)$. If we wish to hold the $i^{th}$ component of $\psi(t)$ fixed, then we must simply set the $i^{th}$ component

of $\frac{d\psi(t)}{dt}$ to zero. Thus the dot product of the $i^{th}$ row of $F(t)$ and the vector $g(t)$ equals zero. If several components of $\psi(t)$ are constrained, then let us stack up all the corresponding rows of $F(t)$ to form a matrix $R_0(t)$. Since the matrix product of $R_0(t)$ and $g(t)$ is a zero vector, we can say that $g(t)$ lies in the null space of $R_0(t)$. Thus during the optimization procedure, the vector $g(t)$ must be constrained to lie in this null space, which is a function of the elements of $\psi(t)$. Chan points out that a nontrivial $g(t)$ satisfying this constraint condition will exist if the number of $\psi$ entries held constant is less than the dimension of $g(t)$.

The next step in the optimization procedure is to express the derivative of the objective function $f(t)$ in terms of $g(t)$. Using the chain rule, and (8.4):

$$\begin{aligned} \frac{df}{dt} &= \sum_{i=1}^{q} \frac{df}{dP_i} \frac{dP_i}{dt} \\ &= \sum_{i=1}^{q} \frac{df}{dP_i} G_i(t)\, P_i(t) \end{aligned} \tag{8.8}$$

Now, by stacking the elements of the $G_i$ matrices as in (8.6), we can define the gradient vector $\xi$ as a linear function of $g(t)$:

$$\frac{df}{dt} = \xi'(t)\, g(t) \tag{8.9}$$

We would like to select the vector $g(t)$ in the negative $\xi(t)$ direction, so that $\frac{df}{dt}$ will be as negative as possible. However, keep in mind that $g(t)$ must also satisfy the null space constraint described above. Thus, if we choose $g(t)$ to be a unit magnitude vector indicating the direction in which the optimization should proceed while satisfying the constraint, then:

$$g(t) = \frac{-\xi_R(t)}{\|\xi_R(t)\|} \tag{8.10}$$

where $\xi_R(t)$ is the projection of $\xi(t)$ onto the null space of $R_0$. As explained in Chan [17], $\xi_R(t)$ can be found by computing:

$$\xi_R(t) = X\,(X'\,X)^{-1}X'\,\xi(t) \tag{8.11}$$

where $X$ is a matrix formed from a set of column vectors which form a basis for the null space of $R_0$.

In order to create an algorithm that will implement the optimization procedure as described above, we must divide the continuous parameter $t$ (call it 'time') into discrete steps of length $h$. Thus the optimization algorithm will involve a series of computations that produces a new transformed structure at time $t+h$ from the transformed structure at time $t$. This process can be repeated until the value $f(t+h)$ of the objective function for the new structure is as small as we like, or until no further significant improvement seems likely.

So for a given structure at time $t$, we can perform all the computations involved in (8.6)-(8.11). The resulting vector $g(t)$ is used to update the transformation matrices by integrating (8.4). Chan uses the simple Euler integration formula to form a tentative $\hat{P}_i$ for the next time instant $t+h$:

$$\hat{P}_i(t+h) = P_i(t) + h\,G_i(t)\,P_i(t) \quad \text{for } 0 \le i \le q \tag{8.12}$$

where $h$ is the integration step size. The reason that this choice is only tentative is that the new structure formed with the transformations $\hat{P}_i(t+h)$ would not in general satisfy the scaling constraints of the original structure. We must include

some scaling operation in order that the structure resulting from the transformations $P_i(t+h)$ is also scaled as desired. Recall from sections 5.2 and 5.3 that $l_2$ scaling involves the diagonal transformation matrices $S_i$ whose elements are the reciprocal square roots of the diagonal elements of a set of matrices $K_i$. In fact, the matrices $K_i(t+h)$ for the new transformed structure can be related to the matrices $K_i(0)$ of the original scaled structure by:

$$\hat{K}_i(t+h) = \hat{P}_i(t+h)\, K_i(0)\, \hat{P}_i'(t+h) \qquad \text{for } 1 \le i \le q \tag{8.13}$$

Note that the diagonal elements of $K_i(0)$ are all unity since we have assumed our original structure to be scaled. Using (8.13), we can describe the *scaling* transformations (5.9) that would have to be applied to the structure resulting from the transformations $\hat{P}_i$ in order to scale it. In particular, the $j^{th}$ diagonal element of $S_i$ would be the reciprocal square root of the $j^{th}$ diagonal element of $K_i$. The diagonal transformation matrices $S_i$ can be combined with the tentative transformation matrices $\hat{P}_i$ to form the *scaled* transformation matrices $P_i(t+h)$:

$$P_i(t+h) = S_i(t+h)\, \hat{P}_i(t+h) \tag{8.14}$$

Thus the structure formed by transforming with the matrices $P_i$ above will have corresponding $K_i$ matrices whose diagonal elements are all unity.

Using the transformations in (8.14), we can compute the new modified state space matrices $\Psi_i(t+h)$ with (8.3). Note that the $\Psi_i$ matrices of the new structure are always computed using by applying the updated transformations of

(8.14) to the $\Psi_i$ matrices of the original structure. In other words, the structure is not formed by updating the $\Psi_i$ matrices of the previous time step. This method was used to keep the effects of numerical inaccuracy to a minimum. Even with the method currently in use, we must consider the fact that the Euler integration of (8.12) is only an approximation to (8.4). Thus, after computing the new $\Psi_i$ matrices, we must check that the constrained entries in each matrix have not changed, that is, we must check to determine whether the errors in the constrained entries are less than some preset tolerance. If these errors were too large, then one approach would be to halve the step size $h$ and repeat the procedure starting with the computation of the tentative transformation updates $\hat{P}_i$ in (8.12). If in fact the errors are small enough, then we should reevaluate the objective function $f(t+h)$. If the resulting value is not smaller than at time $t$, and it need not be due to numerical errors, then we should again use the approach of reducing the step size $h$ and repeating the computations starting with the same updates in (8.12). If the objective function did turn out to be smaller than the value at time $t$, then the optimization procedure can continue for the next time $t+2h$, starting with the original formation of the vector $\psi(t)$ in (8.6).

The overall algorithm can be summarized as follows:

(1) Initialize the procedure with $\Psi_1, \Psi_2, \cdots, \Psi_q$ as $\Psi_1(0), \Psi_2(0), \cdots, \Psi_q(0)$ and compute $K_i(0)$ as described in Chapter 5. Evaluate the objective function, and set all $P_i(0)$ to be identity matrices. Initialize $h$ to 1.

(2) Determine the matrix $F$ and the constraint submatrix $R_0$ as defined in (8.5) - (8.7).

(3) Find a set of basis (column) vectors $\chi_i$ for the null space of $R_0$ and form them into the matrix $X$:

$$X = \begin{bmatrix} \chi_1 & \chi_2 & \chi_3 & \cdots \end{bmatrix} \tag{8.15}$$

(4) Express the derivative of the objective function as a function of $g(t)$, that is, find $\xi(t)$ as defined in (8.9). Find its projection onto the range space of $X$ using (8.11).

(5) Evaluate $g(t)$ using (8.10).

(6) Compute a tentative set of matrices $\hat{P}_i(t+h)$ by Euler integration (8.12), and evaluate the corresponding $\hat{K}_i$ matrices in (8.13).

(7) Scale the $\hat{P}_i(t+h)$ matrices using (8.14) and evaluate the new (scaled) modified state space matrices $\Psi_i(t+h)$.

(8) Check for errors in the constrained coefficients of $\Psi_i(t)$. If any, halve $h$ and return to step 6.

(9) Recompute the objective function $f$. If it has increased, halve $h$ and return to step 6. Otherwise, return to step 2 unless no further improvement is desired.

## §8.3 The Minimization of Roundoff Noise Effects in Compensators

Chan [17,101] applied the general procedure outlined in section 8.2 to the constrained optimization of filter structures for minimimum output roundoff noise variance. In this section we will adapt this technique to the constrained optimization of *compensator* structures for minimum roundoff noise effects. In particular, we will minimize the increase in the performance index $J$ due to roundoff quantiza-

tion noise. In fact, part of our adaptation can also be applied to generalize the technique of Chan for digital filters.

To apply the general technique described in section 8.2 we must specify an objective function $f$, and also express the derivative of $f(t)$ in (8.9) as a function of $g(t)$, or in other words, compute $\xi(t)$. Chan has used an approach similar to that described in section 5.6 to form an objective function. Thus the output noise variance was expressed as a function of the matrices $K_i$, $W_i$, and $\Lambda_i$, which were discussed in section 5.6 for one-level structures. Recall that the these matrices can be found by solving two Lyapunov equations of the same order as the number of unit delays in the filter structure. Thus Chan essentially extended the roundoff noise expression derived by Mullis and Roberts and Hwang to apply to multiple-level filter structures. Chan was then able to define an objective function, and derive an expression for its derivative as necessary in (8.9).

In this section we will adapt Chan's roundoff noise expression for the digital compensator case. Specifically, we will use the context of the modified state space representation, account for the performance of the entire closed-loop system, and also specify the objective function to reflect the increase in the performance index $J$. Thus we will be extending the expression we derived in section 5.6 to the case of multiple-level compensator structures (see (5.44)-(5.49)). We will also show that the expression derived by Chan for the derivative of $f$ applies almost unchanged to the compensator case.

We can extend (5.50) to include multiple precedence levels as follows. Excluding A/D noise, we can rewrite equation (5.33) as: (Tildes represent the quantities of the scaled system.)

$$\tilde{Z} = \tilde{A}\,\tilde{Z}\,\tilde{A}' + \frac{\Delta_r^2}{12}\begin{bmatrix} 0 & 0 \\ 0 & \Omega \end{bmatrix} \tag{8.16}$$

where $\tilde{A}$ is defined in (5.32) and

$$\Omega = \Lambda_q + \Psi_q \Lambda_{q-1} \Psi_q' + \Psi_q \Psi_{q-1} \Lambda_{q-2} \Psi_{q-1}' \Psi_q'$$

$$+ \quad \cdots \quad + \Psi_q \cdots \Psi_2 \Lambda_1 \Psi_2' \cdots \Psi_q'$$

Recall that $\Lambda_i$ is a diagonal matrix whose $j^{th}$ diagonal element represents the number of roundoff noise sources associated with the $j^{th}$ row of $\Psi_i$, $\Delta_r$ is the quantization step size of the quantizers in the structure, $A$ contains the parameters of the closed-loop system, and $Z$ is the steady-state covariance matrix of the plant and compensator states. Also note that the parameter $k_{ad}$ will have no effect on the optimization procedure described below, or on the procedure to be described in section 8.4. Thus it can be set to 1 if desired.

If we replace $\tilde{Z}$ with $T\,Z\,T^{-1}$ as in (5.46), where $T$ is the scaling transformation matrix that relates the original unscaled system of plant and compensator to the scaled system:

$$T = \begin{bmatrix} I_n & 0 \\ 0 & S_q \end{bmatrix} \tag{8.17}$$

then (8.16) can be rewritten:

$$Z = A\,Z\,A' + \frac{\Delta_r^2}{12}\begin{bmatrix} 0 & 0 \\ 0 & \hat{\Omega} \end{bmatrix} \tag{8.18}$$

where $A$ is given in (5.45) and $\hat{\Omega}$ is given below:

$$\hat{\Omega} = \Lambda_q S_q^{-2} + \Psi_q \Lambda_{q-1} S_{q-1}^{-2} \Psi_q' + \Psi_q \Psi_{q-1} \Lambda_{q-2} S_{q-2}^{-2} \Psi_{q-1}' \Psi_q'$$

$$+ \cdots + \Psi_q \cdots \Psi_2 \Lambda_1 S_1^{-2} \Psi_2' \cdots \Psi_q'$$

The expression for $dJ$, the increase in the performance index due to roundoff noise, is given in (5.46), and shown below:

$$dJ = \text{trace} \left\{ \Upsilon Z \right\} \tag{8.19}$$

Using the adjoint Lyapunov equation as described in Appendix B, and as applied in (5.47)-(5.50), we can express $dJ$ as:

$$dJ = \frac{\Delta_r^2}{12} \text{trace} \left\{ W \begin{bmatrix} 0 & 0 \\ 0 & \hat{\Omega} \end{bmatrix} \right\} \tag{8.20}$$

where $W$ is given in (5.48). Defining the lower right-hand (n+1)x(n+1) portion of $W$ to be $W_q$, we can rewrite (8.20):

$$dJ = \frac{\Delta_r^2}{12} \text{trace} \, (\Lambda_q S_q^{-2} W_q$$

$$+ \Lambda_{q-1} S_{q-1}^{-2} \Psi_q' W_q \Psi_q + \Lambda_{q-2} S_{q-2}^{-2} \Psi_{q-1}' \Psi_q' W_q \Psi_q \Psi_{q-1}$$

$$+ \cdots + \Lambda_1 S_1^{-2} \Psi_2' \cdots \Psi_q' W_q \Psi_q \cdots \Psi_2) \tag{8.21}$$

Once we have gotten to this point, the remainder of the development is very similar to the development of Chan [17]. As Chan has done, we can now define the matrices $W_l$. Using a recursive definition,

$$W_i = \Psi_{i+1}' W_{i+1} \Psi_{i+1} \quad \text{for } i=1, \cdots, q-1 \tag{8.22}$$

These matrices are called *noise gain* matrices by Chan, since their diagonal elements reflect the gain from each roundoff noise source variance to the variance of the filter output. For our development, they will represent the gain from each roundoff noise source variance to the increase $dJ$ in the performance index. Applying (8.22), equation (8.21) can be further simplified:

$$dJ = \frac{\Delta_r^2}{12} \sum_{i=1}^{q} \text{trace} \left\{ \Lambda_i S_i W_i \right\} \tag{8.23}$$

or equivalently,

$$dJ = \frac{\Delta_r^2}{12} \sum_{i=1}^{q} \left\{ \sum_j \left[\Lambda_i\right]_{jj} \left[S_i^{-2}\right]_{jj} \left[W_i\right]_{jj} \right\} \tag{8.24}$$

Thus only the diagonal elements of $W_i$ appear in (8.24). Since the diagonal elements of $K_i$ equal the diagonal elements of $S_i^{-2}$, and in fact all equal one for a scaled structure, we can eliminate this term, at least so far as the evaluation of (8.24) is concerned. Since the scale factor $\frac{\Delta_r^2}{12}$ will not affect the minimization process in any way, we can formulate the following objective criterion for the effects of roundoff noise:

$$f = \sum_{i=1}^{q} \left\{ \sum_j \left[\Lambda_i\right]_{jj} \left[W_i\right]_{jj} \right\} \tag{8.25}$$

Now we must turn to the task of expressing the derivative of $f$ as a function of $g(t)$. Chan [17] has shown that the digital filtering $K_i$ and $W_i$ matrices

have the following derivatives:

$$\begin{aligned} \frac{dK_i(t)}{dt} &= G_i(t)\,K_i(t) + K_i(t)\,G_i'(t) \\ \frac{dW_i(t)}{dt} &= -G_i'(t)\,W_i(t) - W_i(t)\,G_i(t) \end{aligned} \tag{8.26}$$

These will apply equally well to the compensator case with its analogous $K_i$ and $W_i$ matrices. Using (8.24)-(8.26) and following the method used in Chan we can write the derivative of the objective function $f$:

$$\frac{df}{dt} = \sum_{i=1}^{q} \left\{ \sum_{j} \left[\Lambda_i\right]_{jj} \left\{ \left( \frac{d}{dt}\left[K_i\right]_{jj} \right) \left[W_i\right]_{jj} + \left[K_i\right]_{jj} \left( \frac{d}{dt}\left[W_i\right]_{jj} \right) \right\} \right\} \tag{8.27}$$

After substituting for the derivatives in (8.27) with the expressions in (8.26) and some manipulation as in Chan [17], we arrive at the following compact expression:

$$\frac{df}{dt} = \sum_{i+1}^{q} \text{trace}\left( M_i'(t)\,G_i(t) \right) \tag{8.28}$$

where

$$\left[M_i(t)\right]_{jk} = 2 \left\{ \left[K_i(t)\right]_{jk} \left[W_i(t)\right]_{jj} \left[\Lambda_i\right]_{jj} - \left[\Lambda_i\right]_{kk} \left[W_i(t)\right]_{jk} \right\} \tag{8.29}$$

The quantity $\xi$ needed in the optimization procedure can easily be obtained from (8.28) and (8.29).

Clearly, the $K_i$ and $W_i$ matrices in (8.29) are defined differently from those derived for digital filters. Other than this, there are two external differences between our expression in (8.29) and the original expression derived by Chan. First, the lack of the factor $\left[K_i\right]_{kk}$ in the second term of (8.29) is due to the

fact that all the diagonal elements of $K_i$ are unity; recall that we assumed that our original structure was scaled. This is largely a procedural difference between Chan's derivation and our own — it terms of the optimization, it makes no difference. The second difference in this expression is the presence of the $\Lambda_i$ term. Recall that the $j^{th}$ diagonal entry of $\Lambda_i$ represents the exact number of roundoff noise sources associated with the $j^{th}$ row of $\Psi_i$. During the optimization procedure, any unconstrained unity or zero entries in these $\Psi$ matrices will in general become non-zero and non-unity. Thus these new sources must also be included in the $\Lambda_i$ matrices at the beginning of the optimization procedure. Inclusion of the $\Lambda_i$ terms allows us to consider all possible structures. The assumption made by Chan, that the $\Lambda_i$ matrices can be taken to be identities or proportional to identities, can often be in error, especially for structures with multiple precedence levels and few coefficients. The result will be only an approximate optimized structure. Our inclusion of the $\Lambda_i$ terms can be easily be incorporated into the digital filtering optimization results of Chan. This is one example where our results can be applied for digital filters.

With the optimization procedure derived by Chan and the correct initial conditions, a structure identical to the minimum roundoff noise filter structure derived in closed form by Mullis and Roberts can be found. Similarly, using the adapted optimization procedure described above and the correct initial conditions, we can also duplicate the minimum roundoff noise compensator structure that we derived in section 5.6. To achieve this result for compensator structures, we must allow all the coefficients (except the next-to-last column of $\Psi_1$) of a one-level initial

scaled structure to vary. Thus all the diagonal entries of the matrices $\Lambda_i$ must be set to $n$+1. Similarly, by allowing only 2 by 2 diagonal blocks of coefficients, plus the last row and column (input and output coefficients) of $\Psi_1$ to vary, we can optimize and produce a block optimal structure. In fact this procedure was used to generate the (one-level) block optimal F8 compensator structure studied in Chapters 5 and 6.

The optimization procedure was also applied to the (scaled) two-level parallel F8 compensator structure composed of direct form II sections designated as (c) in Chapters 5 and 6. Its modified state space (before optimization) is shown below:

$$\Psi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ c & c & c & c & c & c \end{bmatrix} \tag{8.30}$$

$$\Psi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ c & c & 0 & 0 & 0 & 0 & 0 & c \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & c & 0 & 0 & 0 & c \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & c & c & 0 & c \end{bmatrix}$$

where each entry $c$ represents a coefficient. Two extra coefficients were added

by allowing two other entries in $\Psi_1$ to vary, the (5,5) and (5,6) entries. Thus there will be 17 coefficients total in the optimized structure. For this example, the matrices $\Lambda_1$ and $\Lambda_2$ will be:

$$\begin{aligned} \Lambda_1 &= \text{diagonal}\,[0 \quad 3 \quad 0 \quad 3 \quad 2 \quad 3] \\ \Lambda_2 &= \text{diagonal}\,[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 6] \end{aligned} \tag{8.31}$$

Before optimization, the scaled coefficients values ranged from 10.48 to 0.073. Figure 8-1 shows the range of coefficients values and the resulting number of signal bits necessary to hold the value of $dJ$ due to roundoff to 5% (as in Chapter 5) after each iteration of the optimization process:

| Iteration Number | Number Of Bits | Coefficient Range |
|---|---|---|
| 0 | 10.46 | 10.48 - 0.073 |
| 1 | 8.745 | 3.2 - 0.108 |
| 2 | 8.057 | 1.46 - 0.108 |
| 3 | 8.2 | " |
| 4 | 8.086 | " |
| 5 | 8.06 | " |
| 6 | 8.056 | " |
| 7 | 8.057 | " |
| 8 | 8.056 | " |
| 9 | 8.055 | " |

Figure 8-1: *Roundoff Noise Optimization Results*

Without including the 2 extra coefficients, which alters $\Lambda_1$ and increases the apparent required wordlength of the initial structure to 10.46, the number of bits needed (see figure 5-8) was 10.18. Thus the true improvement resulting from the optimization was 2.12 bits. This is quite impressive, since it was attained basically in only two iterations and is quite close to the block optimal value of 7.88 bits, which requires 25 coefficients. In fact, it is identical to the performance of the 17 coefficient parallel structure (b). We note that iterations 3, 4,

5, and 7 involved halving the integration step size due to increases in $dJ$ over the current least value, and that the value after iteration 9 was actually lower than the value after iteration 2, but not appreciably. In the digital filter examples treated by Chan in [17,101], typically only 5 to 8 iterations were required to achieve the full benefits of the optimization procedure. The block optimal compensator structure computed via this optimization procedure took 5 iterations to reach the approximate minimum wordlength.

A byproduct of the optimization procedure for the figure 8-1 example was a reduction in the maximum coefficient value. Instead of needing 4 integer bits to represent the largest coefficient (see Chapter 6), the optimized structure required only one, again an impressive savings in wordlength. Intuitively, this savings may exist for any increase in the number of coefficients in a structure. This point needs more investigation.

## §8.4 The Minimization of Coefficient Wordlength in Compensators

In this section we will develop an objective function for the minimization of coefficient rounding effects on the performance index $J$. Basically, we will use the *MSWL* expression as presented in Chapter 6. The optimization could just as well be carried out for the *SWL* estimate, but the *MSWL* is simpler to compute and still tightly related to the more accurate *SWL* value. This objective function is *quite* different from the one developed in Chan [17], since it is based on $J$, and hence involves second-order sensitivities. Again, as with the *SWL* and *MSWL* derivations, this development will be useful in digital filtering for filters that are designed by optimizing some scalar differentiable criterion.

Instead of minimizing the actual *MSWL* value, we will copy the approach of

the previous section; rather than minimizing the actual required wordlength, we will minimize the expected value of $dJ$. Of course, for the analysis of finite wordlength coefficient effects, this expected value is over an ensemble of structures — it is not a time-average as in the roundoff noise case. Reviewing the results of Chapter 6, $E(dJ)$ can be written:

$$E(dJ) = \frac{\Delta^2}{24} \sum_{m=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_m^2} \right|_\infty \right) \tag{8.32}$$

where $N$ is the number of non-zero, non-unity, and non-power-of-two coefficients in the structure. Thus we can drop the scale factor as we did with the roundoff noise objective function to form a new $f$:

$$f = \sum_{m=1}^{N} \left( \left. \frac{\partial^2 J}{\partial c_m^2} \right|_\infty \right) \tag{8.33}$$

where

$$\frac{\partial^2 J}{\partial c_m^2} = \text{trace} \left( \Upsilon \frac{\partial^2 \tilde{Z}}{\partial c_m^2} \right) \tag{8.34}$$

$$\tilde{Z} = \tilde{A}\, \tilde{Z}\, \tilde{A}' + \begin{bmatrix} \Theta_1 & 0 \\ 0 & \left( \tilde{\Psi}_{12} \Theta_2 \tilde{\Psi}_{12}' \right) \end{bmatrix} \tag{8.35}$$

and $\Upsilon$ contains the performance index weighting matrices as shown in (5.34). The tilde again refers to the parameters of the scaled system. We can also write the expressions (8.34) and (8.35) for the compensator before it is scaled, resulting in:

$$\frac{\partial^2 J}{\partial a_m^2} = \text{trace}\left(\Upsilon \frac{\partial^2 Z}{\partial a_m^2}\right) \tag{8.36}$$

$$Z = A\, Z\, A' + \begin{bmatrix} \Theta_1 & 0 \\ 0 & \left(\Psi_{12}\Theta_2\Psi_{12}'\right) \end{bmatrix} \tag{8.37}$$

where $a_m$ represents a coefficient of the unscaled structure. As with the approach for roundoff noise minimization discussed in section 8.3, we would like to express $f$ as a function of the unscaled parameters and the scaling matrices $S_i$. This is necessary for the computation of the derivative of $f$; even though the original structure selected for the optimization will be scaled, and its $S_i$ matrices will be identities, they do affect the derivative of $f$.

The terms $\frac{\partial^2 J}{\partial c_m^2}$ can be related to the terms $\frac{\partial^2 J}{\partial a_m^2}$ as follows. Since $\tilde{\Psi}_i = S_i\, \Psi_i\, S_{i-1}^{-1}$, a scaled coefficient $c_m$ at index $(j,k)$ in the matrix $\tilde{\Psi}_i$ can be related to its unscaled counterpart by:

$$\left[\tilde{\Psi}_i\right]_{jk} = \left[S_i\right]_{jj} \left[\Psi_i\right]_{jk} \left[\left(S_{i-1}\right)^{-1}\right]_{kk} \tag{8.38}$$

Since $c_m$ is thus a multiple of $a_m$, we can write:

$$\frac{\partial^2 J}{\partial c_m^2} = \frac{\partial^2 J}{\partial a_m^2} \left(\frac{\left[S_{i-1}\right]_{kk}}{\left[S_i\right]_{jj}}\right)^2 \tag{8.39}$$

We can express this relationship compactly for all the coefficients in level $i$ as:

$$Y_2(\tilde{\Psi}_i) = S_i^{-2}\, Y_2(\Psi_i)\, S_{i-1}^2 \qquad (8.40)$$

where $Y_2(M)$ is a matrix function whose dimensions match those of its argument matrix M and whose $(j,k)^{th}$ element is $\dfrac{\partial^2 J}{\partial [M]_{jk}^2}$ only if the $(j,k)^{th}$ location in M corresponds to a multiplier coefficient in the transformed structure, and zero otherwise. Recall that all entries in the precedence level matrices of a structure whose ideal values cannot be represented exactly with a finite number of bits are multiplier coefficients. Thus all zero, unity, and power-of-two entries would not be considered to be multiplier coefficients.

To compute the derivative of $f$ with respect to $t$, we will need to determine the relationship of the second-order sensitivities of $J$ with respect to the coefficients of the transformed structure to the second-order sensitivities of $J$ with respect to the untransformed coefficients. The general transformation matrices $P_i$ in (8.1) and (8.2) are not diagonal, so the simple expression in (8.40) cannot be used. In fact, for the coefficient $c_m$ in the $i^{th}$ level of the transformed structure, the term $\dfrac{\partial^2 J}{\partial c_m^2}$ will now be related to all the second partials of $J$ with respect to the entries in $\Psi_i$ that correspond to multiplier coefficients in the transformed structure, including the *mixed* second partials. To demonstrate this, the following matrix chain rule can be applied [102]; if $x$ and $y$ are scalars, and $M$ a matrix, then:

$$\frac{\partial y}{\partial x} = \text{trace}\left(\frac{\partial y}{\partial M}\ \frac{\partial M'}{\partial x}\right) \tag{8.41}$$

For this expression, the derivative of a scalar with respect to a matrix M is defined to be the matrix whose $(j,k)^{th}$ element is the derivative of the scalar with respect the $(j,k)^{th}$ element of M. Now let the precedence level matrices associated with the transformed structure be designated with the tilde symbol. By applying the matrix chain rule with $J$ as $y$, the coefficient at index $(j,k)$ in $\tilde{\Psi}_i$ as $x$, and $\Psi_i$ as $M$, then we get:

$$\frac{\partial J}{\partial\left[\tilde{\Psi}_i\right]_{jk}} = \text{trace}\left(\frac{\partial J}{\partial \Psi_i}\ \frac{\partial \Psi_i'}{\partial\left[\tilde{\Psi}_i\right]_{jk}}\right) \tag{8.42}$$

Recall from (8.1) that the relationship between the transformed and untransformed precedence level matrices can be written $\tilde{\Psi}_i = P_i\ \Psi_i\ (P_{i-1})^{-1}$. Thus the second term in the trace of (8.42) can be written:

$$\frac{\partial \Psi_i'}{\partial\left[\tilde{\Psi}_i\right]_{jk}} = P_{i-1}'\ E_{kj}\ \left(P_i'\right)^{-1} \tag{8.43}$$

Note that equation (8.42) seems to imply that the derivative $\frac{\partial J}{\partial\left[\tilde{\Psi}_i\right]_{jk}}$ is a function of the matrix $\frac{\partial J}{\partial \Psi_i}$, which involves derivatives with respect to *all* the entries of $\Psi_i$, not just the few coefficient entries. This would imply a tremendous computational load, especially when second derivatives were considered — for a seventh-order compensator with two precedence levels and 7 intermediate nodes,

we would have to compute mixed second partials with respect to all the entries in each level, or $49^2+49^2$ second derivatives. This number would be independent of the number of actual multipliers in the structure. Even though this computation need only be performed once at the start of the optimization process, it would involve far too much computation time. Fortunately, it is not necessary to compute all the derivatives above. In fact, since the matrices $P_i$ are constrained not to vary certain fixed entries in the $\Psi_i$ matrices, the matrix in equation (8.43) will have a special property; it will have zero entries in exactly those locations which will eliminate the dependence of (8.42) on derivatives with respect to $\Psi_i$ entries which are not in the same locations as the multiplier coefficients of the transformed structure. In other words, the derivatives of $J$ with respect to the multiplier coefficients in the transformed structure will be functions only of the derivatives of $J$ with respect to the $\Psi_i$ entries which are in the same exact locations as those multiplier coefficients. To reflect this fact, the term $\frac{\partial J}{\partial \Psi_i}$ in (8.42) should be replaced by $Y(\Psi_i)$, where $Y(M)$ is a matrix function whose dimensions match those of its argument matrix $M$ and whose $(j,k)^{th}$ element is $\frac{\partial J}{\partial [M]_{jk}}$ only if the $(j,k)^{th}$ location in M corresponds to a multiplier coefficient (non-zero, non-unity, etcetera) in the transformed structure, and zero otherwise. Note that this definition of $Y(M)$ is analogous the the definition of $Y_2(M)$ in (8.42). Thus we can rewrite (8.42):

$$\frac{\partial J}{\partial \left[\Psi_i\right]_{jk}} = \text{trace}\left(Y(\Psi_i)\ P_{i-1}'\ E_{kj}\ \left(P_i'\right)^{-1}\right) \tag{8.44}$$

To relate the second derivatives of $J$ with respect to the coefficients of the transformed and untransformed structures, let us take the derivative of (8.44):

$$\frac{\partial^2 J}{\partial \left[\Psi_i\right]^2_{jk}} = \text{trace}\left\{P_{i-1}'\,E_{kj}\ \left(P_i'\right)^{-1}\frac{\partial}{\partial \left[\Psi_i\right]_{jk}}\ \left(Y(\Psi_i)\right)\right\} \tag{8.45}$$

Inside the trace expression above, the matrix chain rule (8.41) can be applied to each non-zero element of the derivative of $Y(\Psi_i)$. For example, if the $(r,s)$ entry of $\Psi_i$ is also a multiplier coefficient, then:

$$\frac{\partial}{\partial \left[\Psi_i\right]_{jk}}\ \left(\left[Y(\Psi_i)\right]_{rs}\right) =$$

$$\text{trace}\left\{P_{i-1}'E_{kj}\left(P_i'\right)^{-1}\frac{\partial}{\partial \Psi_i}\left(\frac{\partial J}{\partial \left[\Psi_i\right]_{rs}}\right)\right\} \tag{8.46}$$

We will define this trace to be $\left[B_i\right]_{rs}$. Interchanging the order of differentiation, and applying the same reasoning that eliminated the extra derivative terms in (8.42), we can express (8.46) as follows:

$$\left[B_i\right]_{rs} = \text{trace}\left\{P_{i-1}'E_{kj}\left(P_i'\right)^{-1}\frac{\partial}{\partial \left[\Psi_i\right]_{rs}}\left(Y(\Psi_i)\right)\right\} \tag{8.47}$$

Note the presence of the mixed second partial derivatives of $J$ in (8.47). Let us

define the matrix $B_i$ to have non-zero entries $\left[B_i\right]_{rs}$ as described in (8.47) where (r,s) is the location of a multiplier coefficient in the transformed structure, and zero otherwise. With this definition, (8.45) can be rewritten:

$$\frac{\partial^2 J}{\partial \left[\tilde{\Psi}_i\right]^2_{jk}} = \text{trace} \left\{ P_{i-1}{}' E_{kj} \left(P_i{}'\right)^{-1} B_i \right\} \tag{8.48}$$

Thus with (8.47) and (8.48), we have now fully described the relationship between the second partial derivatives of $J$ with respect to the coefficients of the transformed structure and the mixed second partials of $J$ with respect to the corresponding coefficients in the untransformed structure.

We can include scaling in this formulation by applying the results derived in (8.38)-(8.40) to the transformed structure. Thus, the complete expression for the objective function $f(t)$ will be:

$$f = \sum_{i=1}^{q} s_i^{-2} Y_2(\tilde{\Psi}_i) s_{i-1}^2 \tag{8.49}$$

where

$$\left[Y_2(\tilde{\Psi}_i)\right]_{jk} = \text{trace} \left\{ P_{i-1}{}' E_{kj} \left(P_i{}'\right)^{-1} B_i \right\} \tag{8.50}$$

and $\tilde{\Psi}_i$ now represents the coefficient parameters *after* transformation but *before* scaling, and $B_i$ is a function of all the mixed second partials of $J$ with respect to the entries in $\Psi_i$ that correspond to coefficients in the transformed scaled structure. Given that the transformed scaled structure will have $N$ coefficients, the advantage to the above formulation of $f$ is that the $N^2$ mixed second-order

coefficient sensitivites need only be computed once, at the start of the optimization procedure. The $N^2$ Lyapunov equations that will have to be solved for these sensitivities cannot be simplified through any application of the adjoint Lyapunov operator described in Appendix B. However, the entire set of equations will have the form (6.21) as described in section 6.4, and can be simplified in the same manner as the computations involved in (6.21). Specifically, the first step of the Lyapunov solution method can be bypassed for all $N^2$ equations. As before, this saves at least 75% of the total computation time involved in such solutions.

Now that we have formulated an expression for $f(t)$, we can examine its derivative with respect to the transformation parameter $t$. Following the procedure of (8.8), we must first evaluate the derivatives of $f$ with respect to the matrices $P_i$, and then multiply the resulting $i^{th}$ term by the matrices $G_i P_i$ for all $i$. From (8.49), $\frac{df}{dP_i}$ will involve the derivative of $S_i^{-2}$, which is a matrix composed of the diagonal elements of $K_i$, and the derivative of its inverse. These can be found by applying (8.26) and a simple matrix identity for the derivative of a matrix inverse [102]. The derivative $\frac{df}{dP_i}$ will also involve the derivative $\frac{dY_2(\Psi_i)}{dP_i}$. This term can be computed easily since the expression for $Y_2(\Psi_i)$ in (8.50) and involving $B_i$ in (8.47) is a direct function of the $\Psi_i$ matrices. All the other terms in (8.47) and (8.50) are not dependent on the $\Psi_i$ matrices. The actual formation of $\xi(t)$ in (8.9) from the resulting derivative expressions will be quite tedious, but really is only a matter of bookkeeping. As a whole, the method we have described above is computationally quite efficient. We have not tested the optimi-

zation procedure of section 8.2 in the context of the statistical wordlength-based objective function (8.49) for an actual example.

## §8.5 Criteria For Selecting Unconstrained Coefficients

As stated by Chan [101], one of the major open issues concerning this optimization procedure relates to selecting which entries in the $\Psi_i$ matrices will be constrained. For the optimization of parallel or cascade compensator structures composed of second-order sections, we have formulated some general guidelines that seem appropriate. As will be shown, these guidelines can be applied equally well to the digital filtering case.

For the optimization of roundoff noise effects, the block optimal form of Mullis and Roberts, and Hwang still tends to have too many coefficients, as compared with structures of nearly the same performance. However, it is possible to use block optimal sections combined with direct form II sections, thereby saving several coefficients. In order to select the section that should be converted to a block optimal section, we must examine the objective function $f$ in (8.25). Recall that $f$ depends only on the diagonal elements of the matrices $\Lambda_i$ and $W_i$. The matrices $\Lambda_i$ reflect the number of roundoff sources that are associated with the rows of the matrices $\Psi_i$, and the diagonal elements of the matrices $W_i$ contain the gains from the variances of the intermediate nodes $r_i$ to the performance index $J$. For a parallel direct form II structure (see (3.25) and figure 3-8), which has two levels, the diagonal elements of $W_1$ will be pairwise associated with the specific second-order sections. Since we know the weights $\left[\Lambda_i\right]_{jj}$, the relative

diagonal values of $W_i$ will indicate which sections in the structure contribute the most to the objective function $f$. The matrix $W_2$ for a parallel direct form II structure will not be important to this consideration, since $\Psi_2$ contains multiplier coefficients and hence roundoff sources that only affect the output node. Recall from (8.1) that this node cannot be altered by the optimization procedure.

Let us consider the example treated in section 8.3. For this structure, the parallel structure (c), the diagonal values of $W_1$ were as follows:

$$\left[W_1\right]_{jj} \text{ for } 1 \le j \le 6 = \left\{1.71,\ 7.32,\ .092,\ .264,\ 342,\ 465\right\} \qquad (8.51)$$

Since the diagonal values of $W_1$ are pairwise associated with the three second-order sections of this example, we can easily identify the third section as the trouble spot — the third pair of values (342,465) is clearly the largest, given the weights $\Lambda_1$ in (8.31). This fact justifies the specific location of the two extra coefficients chosen to be varied. In fact, if we had allowed the section to be truly a block optimal section, it would have required three extra coefficients and not two. However, in this example there are indications that the performance with two could be quite excellent — hence one should not automatically go to a block optimal section. Certainly, this point requires further investigation.

When optimizing only a portion of a structure as discussed here, it is necessary to know the performance level that would result for the block optimal case, so that one can judge the effectiveness of using fewer coefficients. This value can be found using this same optimization procedure, but with more unconstrained $\Psi$ entries (more multiplier coefficients). Note that this approach to determining which section of a structure to optimize can also be adapted to include

cascade structures. We should also mention that the above guidelines will of course not be too effective if the diagonal elements of $W_1$ tend to be similar in magnitude.

A similar guideline may be used when minimizing coefficient wordlength. As mentioned in Chapter 6, by computing the *MSWL* or *SWL*, we have already computed the second partials of $J$ with respect to the coefficients in the structure. Furthermore, the *SWL* computation will also produce the *mixed* second partials of $J$. It is precisely these sensitivities that we need to produce $f$ in (8.49). We would simply have to compute the *SWL* of the original structure $\{\Psi_i(0)\}$, and save the sensitivities. If any of the second-order sensitivities $\frac{\partial^2 J}{\partial a_m^2}$ of the original structure are particularly large as compared to any others, then the second-order section in which those coefficients reside would be a likely candidate for optimization. In particular, any zero or unity entries in the portion of the $\Psi_i$ matrix corresponding to that section should be unconstrained in the optimization procedure. Such a section, when optimized, will have the same form of modified state space representation as a block optimal section, but it will be optimal with respect to a different criterion.

Although the criteria presented above by no means fully answer the question of which $\Psi_i$ entries to constrain, they do provide an important guide in situations where performance *and* minimal numbers of coefficient multiplies are important.

In one sense, the constraint issue is part of a larger topic; the selection of an initial structure from which to optimize. One property of the iterative con-

strained optimization procedure described in this chapter is that the number of precedence levels is fixed during the optimization. Therefore, optimizing a two-level structure for some objective function does not tell us whether an extra level will significantly improve performance, or if one less level can be used without degrading performance. In general more levels provide more degrees of freedom for the optimization, but of course this will depend on the number of constrained coefficients and their locations in the $\Psi_i$ matrices. For now, these questions must be dealt with by trying different initial structures, with different numbers of levels. Further work is needed in this area, both for the synthesis of digital filter structures, and for the synthesis of digital compensator structures.

# Chapter 9: Summary, Conclusions, and Future Efforts

## §9.1 Summary and Conclusions

In this section we will outline the basic points developed in this thesis. We will especially stress the difference between the issues as they relate to digital compensators as opposed to digital filters.

Many elegant mathematical solutions exist for control problems. Often, the resulting compensators are directly implemented on large-scale computer systems, where speed and accuracy are assured, and cost not critical. The issues involved in the implementation of such compensators on small-scale digital systems have not received the attention they deserve. For these applications, the finite memory, relatively slow speeds, and the expense of the hardware *must* be considered in the overall design process. Fortunately, these very issues have been examined in the context of digital signal processing, and a great many useful results exist. Our approach was to *use*, *adapt*, and *extend* these results to digital feedback compensators. This development is essentially the contribution of the thesis. In several situations, however, we have extended these results to the point where they also constitute a useful extension for digital filtering applications. These extensions will also be pointed out in this summary.

The steady-state LQG control and estimation problem was selected as a basic framework for several reasons. First, this type of controller has been shown to have desirable performance properties in terms of its robustness, multivariate formulation, optimal nature, and so forth. Second, the LQG problem has received a great deal of attention in the recent literature, and is being increasingly applied to real systems. Third, the LQG problem has an explicit scalar objec-

tive function, which can be adopted as a performance metric against which the degradation due to finite wordlength effects can be measured. It is not necessary to choose such a performance measure or even the LQG problem at all. However, this choice allows us to develop results in a concrete setting. Finally, using the LQG control framework, we can bring out all the issues we wish to raise, and this can in fact be done using single-input single-output systems. As we will discuss, extensions to the multiple-input multiple-output case are straightforward, although the issue of multiple-input multiple-output structures remains largely unexplored.

In Chapter 2 we presented the assumptions, problem statement, and solution method involved in an LQG system, and raised a key point. The calculations involved in producing the compensator output and state values require a finite amount of time $t_c$. This time must be accounted for in the LQG design procedure. Two implications arise: 1) the sampling period must be greater than $t_c$, and 2) the compensator output at a given sample time can only depend on *past* compensator state and input values. However, if $T >> t_c$, we must not constrain the system to wait a full $T$ seconds for its control update. It *should* only have to wait $t_c$ seconds. Hence, we presented the LQG solution method and sample-skew idea given in Kwakernaak and Sivan [1].

Once such an *ideal* compensator is designed, it must be implemented in finite-precision hardware. In Chapter 3 we presented the concept of a *structure* as defined for digital filters, and the notation introduced by Chan for representing such structures. The concept of an accurate notation for reflecting the arithmetic and quantization operations in a structure and the inherent precedence of these

operations is critical; although all structures have the same transfer function and same performance as the ideal compensator *under infinite precision*, they will in fact all differ, in general, under finite-precision arithmetic. For control applications, two points were stressed. First, a state space is insufficient to represent all possible structures. In fact, it can represent only that class of structures possessing one precedence level. Second, and more important, the notation developed by Chan for filter structures is not quite suitable for representing compensator structures — in fact, the concept of a structure is slightly different in control applications. In digital filtering, the calculation time necessary to compute the next filter output from the current filter states is ignored, since it only represents series delay time. Whether the filtered data emerges 0.1 seconds after its input, or 0.15 seconds, is really of no concern, as long as the data rate is high enough. However, this delay *must* be included in any compensator structure, since this structure is embedded in a feedback loop. If one considers this delay as part of the plant (that is, as a series delay following the compensator), then this effectively raises the dimension of the plant and of any compensator designed via the LQG approach. On the other hand, including the delay *as part of the compensator* accurately describes the operation of the compensator since every unit delay corresponds to a storage register, and allows us to consider more *general* structures in which the added delay does not appear as a series delay following the compensator. Thus we adapted Chan's notation for compensator structures, and called it the *modified state space representation.* It has all the advantages of Chan's notation for digital filters, and furthermore includes *all* the calculation delays that exist in the compensator. A major implication of this definition of compensator structures is that a delay-canonic structure (one that has a minimal

number of delays) for an $n^{th}$-order plant and $n^{th}$-order compensator has $n+1$ unit delay elements, instead of $n$ as in digital filtering. Thus a cascade of direct form I second-order sections, not canonic for digital filters, *is* canonic for digital compensators. In the context of this definition of structures, we presented several classes of structures and pointed out that a straightforward implementation of the ideal compensator equations (called a 'simple' structure) is *not* usually a good choice for steady-state LQG compensators, since it has many more coefficients than nearly every other structure used in digital filtering. Of course, for situations where it was not convenient to compute the parameters of any structures other than the simple structure, such as in adaptive control systems or in any system where the appropriate Ricatti equations must be computed online, the simple structure or a one- or two-level version of this structure (still with many coefficients) must be used.

In Chapter 4 we presented several digital computer architecture concepts as they relate to digital filters and to digital compensators. The basic idea of serialism and parallelism, the degree to which processes run sequentially or concurrently, extends without modification to digital compensators. The intuition that can be gained concerning precedence and maximally-parallel architectures from the Chan notation for digital filters is identical to that gained from the modified state space representation for digital compensators. However, the same cannot be said concerning the application of pipelining to compensators. In fact, the application of pipelining to compensators brings out another point — the interaction between the ideal design process discussed in Chapter 2 and the implementation of the resulting compensator. Basically, the use of pipelining alters a structure so that the number of precedence levels in the structure is reduced, while still pro-

ducing *nearly* the same transfer function. The only difference is the addition of one or more series delay units. Fewer precedence levels means a smaller minimum calculation time and a *faster possible sampling rate*. For digital filters, the extra series delay encountered is of no importance, as discussed above. What is significant is the potential increase in the data rate. However, for compensators, this delay *must* be considered in the design process. If ignored, this delay results in extra negative phase shift and the performance of the control system may deteriorate — it may even become unstable, as demonstrated in Chapter 4. To include the effects of the delay, we can simply increase the order of the plant (with one additional state per unit delay added) and redesign the optimal LQG compensator for the higher sampling rate. The resulting higher-order compensator structure must be able to be pipelined in the same manner as was the original structure. Depending on the application, the pipelined control system with its increased sample rate can have superior performance as compared to the original, slower, non-pipelined system.

In the next three chapters, the effects of finite wordlength in digital compensators were investigated. These effects were divided into three areas: the uncorrelated effects resulting from quantization of the multiplier products (quantization noise, Chapter 5), the correlated effects of these same quantization operations and the overflow nonlinearities in the compensator (limit cycles, Chapter 7), and the effects of quantizing the infinite-precision coefficients of a structure (coefficient quantization, Chapter 6).

The analysis of quantization noise includes an important sub-issue — scaling. Scaling is necessary to match the dynamic range of the signals in the structure to the dynamic range representable with the fixed-point words. Various

types of scaling were described for digital filters, depending on the known characteristics of the compensator input signal; some are more conservative than others (because they assume less is known about the input), thereby resulting in higher noise levels. For digital feedback compensators, two issues were brought out. First, the common LQG set-point configuration makes use of a compensator with *two* inputs, either or both of which can have DC components. This fact would require that the most conservative type of scaling be used ($l_1$ scaling), and would in fact require the use of techniques for scaling multiple-input structures. However, we show that the use of an *alternate* but equivalent set-point configuration can avoid this problem. With the alternate configuration, the compensator has only one input, and this input has no DC component. Thus a less conservative scaling procedure ($l_2$ scaling) can be employed. The *stochastic* scaling method applied equalizes the probability of overflow at every node in the structure. However, this probability depends on the behavior of the *entire* closed-loop system, not the compensator alone (which could be unstable). Thus we have adapted this digital signal processing scaling procedure for use with digital *compensators*.

Once a structure is scaled, we can compute the effect of quantization noise on some objective criterion. For digital filters, we presented the modelling associated with *roundoff* and *sign-magnitude truncation* quantization, and restricted the analysis to the more tractable (and lower noise) case of roundoff. To compute the noise power due to roundoff errors at the output of a digital filter, a Lyapunov equation of order $n$ can be solved, where $n$ is the number of unit delays in the filter. For digital compensators, again, the effect of roundoff errors on the performance index is a *closed-loop* phenomenon. Thus we have *adapted* the

analysis method to include the entire plant and compensator system, as we did for compensator scaling. In addition, for digital signal processing applications, Mullis and Roberts have derived a one-level minimum roundoff noise filter structure. It proved possible to adapt this method to produce a minimum roundoff noise compensator structure. As before, the entire closed-loop system had to be considered.

To test the roundoff effects of different structures for implementing a higher-order compensator, the F8 example was introduced. The results from a roundoff analysis of these structures brought out several points. First, as in digital filtering, the direct form II structure had poorer performance in terms of the increase in $J$ due to roundoff noise than factored forms like the cascade or parallel structures, and as in digital filtering, the pairing and ordering issues associated with cascade structures were significant in determining their performance. As expected, the block optimal minimum roundoff noise compensator structure was better than any of the other structures tested. However, two points were raised that were different for digital compensators as compared to digital filters. First, the pairing issue is further complicated in control compensators due to the presence of many *real* poles. Most digital filters have at most one real pole. However, controllers can frequently have more than one real pole. Thus these poles must be paired if second-order sections are to be used. The same applies to real zeros. Thus even a parallel *compensator* structure brings out the pairing issue, where parallel *filter* structures have no such consideration. Secondly, the default 'simple' structure for digital compensators, not used for filter structures, did perform comparatively well. However, there were two structures with many fewer coefficients that did even better.

The effect of *coefficient* rounding on performance is basically a deterministic one. Given a set of coefficients, we can compute exactly the resulting performance degradation. However, in digital filtering, a statistical approach based on first-order sensitivities has been developed for estimating the coefficient wordlength required to meet some degradation level. Thus it is not necessary to directly evaluate the performance repeatedly until a suitable wordlength is found. We have extended the statistical approach to the LQG compensator, and in so doing, have raised an important point. Because the LQG compensator minimizes the performance index $J$, all first-order sensitivities with respect to the compensator coefficients are zero. Thus second-order sensitivities are necessary to estimate the increase in $J$ due to coefficient rounding, and in fact $J$ can only increase with such rounding. The necessity for second-order terms will be true of *any* parameter optimization problem, for example, sub-optimal control problems like reduced-order compensators. In fact, if a *digital filter* is designed to minimize some differentiable scalar objective function, then a statistical wordlength estimate for this filter using this same objective function must also use second-order sensitivities. This constitutes an extension to the results for the implementation of digital filters.

Other issues concerning coefficient wordlength are raised when we apply the statistical methods developed to the F8 system. First, we have evaluated the structures according to the wordlength required to achieve a specific degradation level. As in digital filters, there was a strong correlation between the low noise and low coefficient sensitivity structures. Again, for digital compensators, we can state that the 'simple' structure performed well, but was still outperformed by the same two structures as in the roundoff analysis. The *SWL* sta-

tistical estimate developed using second-order sensitivities, a new concept, proved to be conservative as is its filtering counterpart based on first-order sensitivities. However, for the five structures requiring the least bits, it was very accurate (0 to 1.4 bits conservative). The *SWL* value was much more conservative for the poorer structures: the direct form II, and the cascade and parallel structures using identical inadvisable pole pairings. *Unlike* the usual digital filter statistical estimate, a second simpler-to-compute estimate was possible, based only on the mean degradation in performance. (This value would be zero for any estimate based on first-order sensitivities.) This *MSWL* estimate was *very* tightly related to the *SWL* value, from .68 to .94 bits lower in all 10 cases, and can thus easily be used for a relative wordlength comparison between several candidate structures or in an optimization algorithm. The major advantage of these two statistical estimates over a deterministic determination of wordlength was *not* in the computation time saved, which was minimal (15% - 30%) for under 20 coefficients and nonexistent for over 20, but in one very important area. Since the estimates were continuous in nature and differentiable, they could be used as the scalar objective function for a structural optimization procedure. In such a procedure based on the statistical estimate, we had to compute all the (mixed) second partial derivatives of $J$ with respect to the $N$ coefficients — but this needed to be done only once for the entire iterative procedure. This point was further developed in Chapter 8.

In the discussion on limit cycles in Chapter 7, we reviewed the methods used in digital filtering for dealing with limit cycles. Although our results in this area were limited, four observations relating to digital compensators were brought out. First, a control system with an open-loop unstable plant, or a plant with an

integrator pole, must of necessity have some sort of low-amplitude limit cycle. The system output will increase from zero until it reaches the lowest quantization level of the output A/D. Only then can control action seek to restore the system to the zero level — but then the process will repeat. This situation is unavoidable since the system is essentially *open-loop* when the magnitude of the output level is less than one A/D quantization level. Second, the global feedback loop around the compensator will change the nature of the limit cycles in the compensator, and can even cause limit cycles. For example, a finite impulse response filter will not exhibit limit cycles, yet a feedback system using a finite impulse response compensator may exhibit limit cycles. Third, the techniques used in filtering for dealing with limit cycles do not often extend to compensators, especially when the plant has an integrator or right-half plane pole. Finally, based on the random rounding and experimental results in the digital signal processing literature, it is not clear whether any limit cycles will exist in LQG systems. The noise driving the system and the noise in the output will tend to quench any limit cycle that may occur. This of course will depend on the intensity of the noise. However, even though limit cycles themselves may be suppressed, other nonlinear effects such as jump discontinuities may occur. Furthermore, the quantization noise in the system is *not* white, and the very presence of correlated noise in the system may cause difficulties. There are few techniques for handling these effects, even for digital filters.

The final topic we treated in the thesis is the iterative constrained optimization of structures. The basis for this technique lies in the work of Chan for filters. However, we can again adapt the algorithm to handle digital compensators. For minimizing roundoff effects, the adaptation was quite similar to that required to

compute the closed-form block optimal one-level minimum roundoff noise structure of Chapter 5. However, for minimizing coefficient roundoff effects, our extension is quite different from the Chan approach, since our statistical estimate is based on *second*-order sensitivities. We demonstrated the optimization technique for roundoff noise effects for several structures, but did not test the changes required to produce a minimum coefficient wordlength structure. Our effort in optimization did bring out two points which extend the optimization technique of Chan *for digital filters also.* First, our technique for the constrained minimization of roundoff noise was more general than that of Chan. We accounted for the exact number and location of roundoff error sources in the structure; Chan uses an approximation to simplify his analysis. This change can easily be incorporated into Chan's filter structure optimization algorithm. Secondly, we pointed out some general approaches to selecting which portion of a compensator structure should be optimized, that is, the portion that will produce the greatest improvement when optimized. These guidelines also apply to the optimization of filter structures.

## §9.2 Future Efforts

Based on our results, there are several extensions that should be mentioned, and also several new issues that we did not address. Let us first consider some of the extensions, both to other performance criteria and to other control or estimation problems.

In principle, our results extend to the consideration of other performance measures, such as gain margin, phase margin, and so forth. However, the details of the derivations and the actual equations will be quite different. For example, the statistical wordlength estimate may be dominated by first-order sensitivities.

However, for the steady-state Kalman filter problem (considered at length by Sripad [13]), our results would be more directly applicable. As in the LQG case, this problem has a simple minimized scalar objective function, the trace of the error covariance matrix. However, since this is not a control problem, but an estimation problem, it will have many of the characteristics of a digital filter. Thus, while a statistical wordlength procedure for the Kalman filter will require the use of second-order sensitivities (like the LQG case), the scaling and roundoff analysis procedures will not depend on any *closed-loop* system behavior (unlike the LQG case). Still, the adaptation of our results and techniques to digital Kalman filter implementations will be fairly straightforward. Of course, the Kalman filter would have to be considered to be a multiple-output compensator (see the discussion below on multiple-input multiple-output systems).

Our efforts can also be easily extended to certain sub-optimal parameter optimization control problems. Both the optimal nature and the closed-loop aspects of the LQG problem are found in these controllers. In fact, if the same $J$ is taken to be the performance measure, all our results apply. The equations will differ only in the fact that, in general, the compensator dimension will be smaller than the plant dimension.

As mentioned above, there are several issues which we did not consider in our work. The first of these involves the nature of the LQG problem. By expressing all the desired performance characteristics of a control system in a single all-encompassing scalar function $J$, there can be some question as to the relevance between the minimization of $J$ and the satisfying of the initial performance objectives. The work of Harvey and Stein [24] mentioned in Chapter 2 is an important step towards solving this problem. What we can state is the following: to the

extent that the index $J$ is relevant to the desired control system performance, our analyses based on increases in $J$ will be relevant to the *relative* performance of an implementation.

Another important issue is the application of our results to multiple-input multiple-output compensators, since there are a great many real-world systems that are multiple-input multiple-output in nature. Given some multiple-input multiple-output structure, our results apply with only a few minor changes. However, the whole question of how one designs multiple-input multiple-output structures is basically unexplored. The modified state space notation is sufficiently flexible to cover the multiple-input multiple-output case, if we simply have input and output vectors, instead of scalars. *Multiple-output* scaling is no problem, since the present technique already scales all the nodes. However, some modifications will be required to implement scaling procedures for *multiple-input* LQG compensators. Certainly we can still compute the variances of all the nodes of the compensator, accounting for the closed-loop nature of the control system, and its driving and measurement noises. Recall that the aim of the stochastic scaling procedure was to equalize the probability of overflow at all the compensator nodes and the compensator input (plant output). However, for multiple-output plants (multiple-input compensators), there is a problem. Figure 9-1 shows a simple double-input compensator. The variances of the two system outputs $y_1$ and $y_2$ will not in general be the same. Thus we cannot equalize the probabilities of overflow at every node and every compensator input. One possible solution is to select only one of the compensator inputs to have the same probability of overflow (after scaling) as all the nodes, and to allow the remaining compensator inputs to have a lower probability of overflow. This can be accomplished by choosing the compensator input

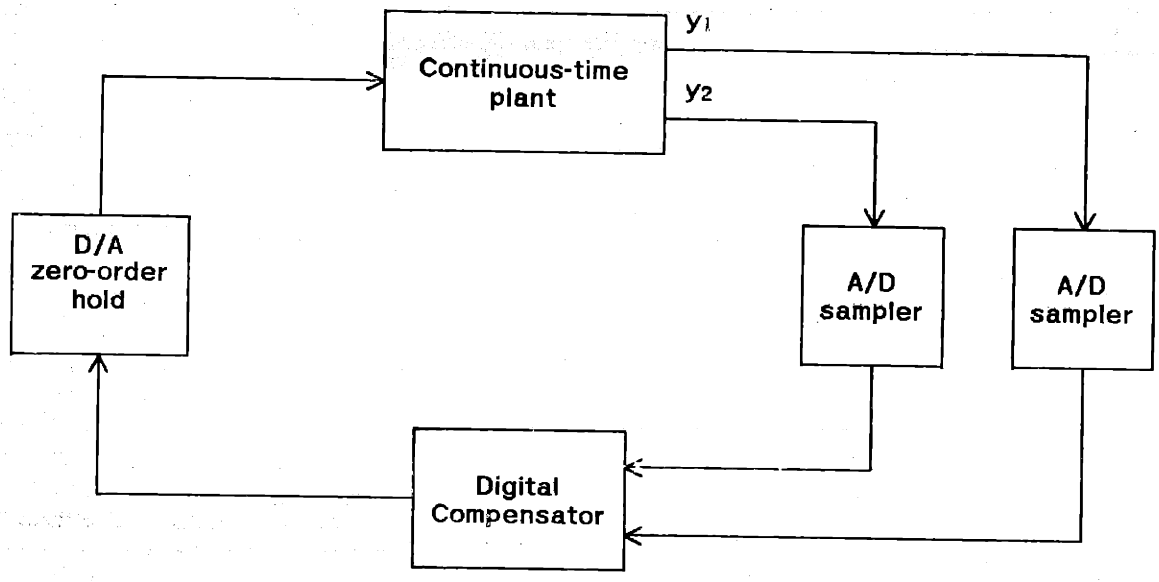



Figure 9-1: Double-Input Compensator Control System

$y_i$ with the largest variance for use in the scaling procedure of Chapter 5. Instead of normalizing $K_q$ in (5.21) and $K_i$ in (5.23) by dividing by the variance of $y$, we will use the variance of $y_i$. However, in equation (5.22), the symbol $y$ must refer to the vector $y$, not $y_i$. Other than these changes, the rest of the compensator scaling procedure basically remains the same. (In the full multiple-input multiple-output scaling procedure, recall that $u$ must also be a vector.) One other point involving scaling should be mentioned. Since each A/D unit has its own scale factor, we must also consider this scaling issue in the multiple-input sense. However, to preserve the overall ideal system performance, all these scale factors must be the same. Again, their choice will depend on the plant output whose combined variance/system transients are the largest.

The question of multi-loop limit cycles does not really further complicate the

limit cycle question. If any effective limit cycle analysis method is found for dealing with single-loop control systems, it should directly extend to the multi-loop case.

Limit cycles themselves may not be an issue for LQG control systems. However, there is a middle ground between white additive quantization noise and a limit cycle oscillation. Jump phenomena and the presence of correlated noise can be very detrimental to control applications. The work of Sripad [13,56] and Parker and Girard [103] on the correlated nature of quantization errors should serve as a foundation for studying such effects.

Another important issue is involved in the constrained optimization of structures presented in Chapter 8. At one level, more work needs to be done in testing and evaluating the minimization of coefficient wordlength. However, on a more fundamental level we have the question of how to select the initial structure. (Recall that the iterative optimization procedure must begin with a specific structure and then apply transformations to it.) The choice of initial structure is important because the iteration procedure cannot change the number of precedence levels in the initial structure. The question of how many precedence levels to use is a very complex one. It is dependent on the number of (unconstrained) coefficients desired, the speed requirements of the application, and the acceptable level of performance degradation. Furthermore, given an initial structure, we do not always know the best way to choose *which* coefficients to constrain. Such considerations are of importance to the optimization of both digital compensator structures and digital filter structures.

Finally, we wish to mention a longer term effort that may become of importance to control engineers. This thesis effort has assumed right from the start

that a fixed-point numerical representation is being used. This implies minimal expense and minimal computation time as compared to floating-point arithmetic computation. However, as the hardware evolves, new systems of arithmetic arise that may be competitive with fixed-point. Particularly, a system called FOCUS [104] has been reported in the literature. The main motivation for FOCUS has been the problems encountered in control and certain other signal processing applications. Specifically, control systems require the most accurate control signals when the system output is close to the desired level (to reduce steady-state error) and less accurate control levels when far from the desired set point. The FOCUS system of numerical representation and arithmetic combines the accuracy advantages of floating point with the hardware simplicity and higher speed of fixed point. Applications of our work on compensator implementation to the FOCUS number system may become quite useful for control systems.

The purpose of this thesis was to expose the fundamental issues involved in the digital implementation of control compensators, and to use, adapt, and extend the techniques of digital signal processing in order to develop methods applicable to control. We believe that our efforts have provided the foundation for an overall methodology for the implementation of compensators.

## Appendix A: F8 Data

This appendix will present the continuous-time F8 model discussed in Chapters 5 and 6, and its discrete-time equivalent. The $G$ and $K$ matrices computed by the procedures mentioned in Chapter 2 are also given. Finally, data defining all 10 candidate structures analyzed in Chapters 5 and 6 and also the optimized structure discussed in Chapter 8 will be presented.

The parameters of the sixth-order single-input single-output continuous-time F8 system are given below, following the notation of Chapter 2:

The A matrix for the continuous-time sixth-order F8 system:

| | | | | | |
|---|---|---|---|---|---|
| -6.696d-01 | 5.7000d-04 | -9.010d+00 | 0.000d+00 | -1.577d+01 | 0.00d+00 |
| 0.000d+00 | -1.3457d-02 | -1.411d+01 | -3.220d+01 | -4.330d-01 | 0.00d+00 |
| 1.000d+00 | -1.2000d-04 | -1.214d+00 | 0.000d+00 | -1.394d-01 | 0.00d+00 |
| 1.000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 | 0.000d+00 | 0.00d+00 |
| 0.000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 | -1.200d+01 | 1.20d+01 |
| 0.000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 | 0.000d+00 | 0.00d+00 |

The B matrix:

| |
|---|
| 0.000d+00 |
| 0.000d+00 |
| 0.000d+00 |
| 0.000d+00 |
| 0.000d+00 |
| 1.000d+00 |

The C matrix:

| | | | | | |
|---|---|---|---|---|---|
| 1.000d+00 | 3.091d-03 | 3.128d+01 | 1.000d+00 | 3.592d+00 | 0.000d+00 |

The $\hat{Q}$ matrix for the state norm:

| | | | | | |
|---|---|---|---|---|---|
| 6.637d+00 | 0.0000d+00 | 0.0000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 |
| 0.000d+00 | 2.6554d-07 | 2.6860d-03 | 0.0000d+00 | 3.085d-04 | 0.000d+00 |
| 0.000d+00 | 2.6860d-03 | 2.7174d+01 | 0.0000d+00 | 3.121d+00 | 0.000d+00 |
| 0.000d+00 | 0.0000d+00 | 0.0000d+00 | 2.7174d+01 | 0.000d+00 | 0.000d+00 |
| 0.000d+00 | 3.0850d-04 | 3.1210d+00 | 0.0000d+00 | 3.585d-01 | 0.000d+00 |
| 0.000d+00 | 0.0000d+00 | 0.0000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 |

The $\hat{R}$ matrix for the control norm:

5.2520d+00

The driving noise covariance $\Xi_1$:

| | | | | | |
|---|---|---|---|---|---|
| 0.000d+00 | 0.0000d+00 | 0.0000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 |
| 0.000d+00 | 0.0000d+00 | 0.0000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 |
| 0.000d+00 | 0.0000d+00 | 0.0000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 |
| 0.000d+00 | 0.0000d+00 | 0.0000d+00 | 0.0000d+00 | 0.000d+00 | 0.000d+00 |
| 0.000d+00 | 0.0000d+00 | 0.0000d+00 | 0.0000d+00 | 1.000d-06 | 0.000d+00 |
| 0.000d+00 | 0.0000d+00 | 0.0000d+00 | 0.0000d+00 | 0.000d+00 | 1.000d-06 |

The measurement noise covariance matrix $\Xi_2$:

1.8441d-03

The discrete-time parameters for the above system sampled at 10 Hertz were computed according to the equations in Chapter 2:

Discrete-time Transition matrix $\Phi$: (Every two rows shown below is actually only one row of the matrix $\Phi$)

| | | |
|---|---|---|
| 8.94158899875840d-01 | 5.93355418621274d-05 | -8.07897769474215d-01 |
| -9.45729891346440d-05 | -8.61683445782787d-01 | -6.35698693844013d-01 |
| -2.22006964581917d-01 | 9.98658866653035d-01 | -1.26236417743289d+00 |
| -3.21783948076557d+00 | 7.07041304940429d-02 | 1.33003345781691d-02 |
| 8.96632406763084d-02 | -8.47007233273550d-06 | 8.45357360609777d-01 |
| 1.55118776302647d-05 | -5.82375346184306d-02 | -2.82235577782270d-02 |
| 9.53225781519899d-02 | 2.93704935200758d-06 | -4.20099682385258d-02 |
| 9.99996866444939d-01 | -5.29748911536677d-02 | -2.33843803649613d-02 |
| 0.00000000000000d+00 | 0.00000000000000d+00 | 0.00000000000000d+00 |
| 0.00000000000000d+00 | 3.01194227548261d-01 | 6.98805772451740d-01 |
| 0.00000000000000d+00 | 0.00000000000000d+00 | 0.00000000000000d+00 |
| 0.00000000000000d+00 | 0.00000000000000d+00 | 1.00000000000000d+00 |

Input matrix $\Gamma$:

| |
|---|
| -2.33843803649613d-02 |
| 1.22296689664596d-05 |
| -8.09744421678703d-04 |
| -6.19868380965549d-04 |
| 4.17661813028933d-02 |
| 9.99999956738716d-02 |

State weighting matrix Q:

| | | |
|---|---|---|
| 6.19891054286165d+00 | 3.01509191580769d-04 | -1.49453104023977d+00 |
| 1.31875722331738d+00 | -3.31370870518672d+00 | -1.39174639302283d+00 |
| 3.01509191580769d-04 | 2.47848377796048d-07 | 2.26275446201938d-03 |
| 2.60488981153273d-05 | -1.83146582461534d-05 | 3.31476135012857d-05 |
| -1.49453104023977d+00 | 2.26275446201938d-03 | 2.49886408802715d+01 |
| -3.91315114446321d-01 | 3.00983380027126d+00 | 1.93726081288867d+00 |
| 1.31875722331738d+00 | 2.60488981153273d-05 | -3.91315114446321d-01 |
| 2.71739588981900d+01 | -5.29472742893936d-01 | -1.68559768886550d-01 |
| -3.31370870518672d+00 | -1.83146582461534d-05 | 3.00983380027126d+00 |
| -5.29472742893936d-01 | 2.37775667424236d+00 | 1.15440114387846d+00 |
| -1.39174639302283d+00 | 3.31476135012857d-05 | 1.93726081288867d+00 |
| -1.68559768886550d-01 | 1.15440114387846d+00 | 6.65122863060225d-01 |

Cross-weighting matrix M:

-3.53295371817073d-02
1.98100358396956d-06
6.49143025809568d-02
-3.50993906625930d-03
3.18190737871274d-02
2.03591671654107d-02

Control weighting matrix R:

5.25266793029727d+00

Output matrix L:

| | | |
|---|---|---|
| 1.00000000000000d+00 | 3.09100000000000d-03 | 3.12800000000000d+01 |
| 1.00000000000000d+00 | 3.59200000000000d+00 | 0.00000000000000d+00 |

State driving noise covariance matrix $\theta_1$:

| | | |
|---|---|---|
| 4.33676915174706d-08 | -1.04991167649513d-09 | 2.00314796017624d-09 |
| 1.67658416889601d-09 | -3.93401117854579d-08 | -2.33843803650305d-08 |
| -1.04991167649513d-09 | 6.13454985323572d-11 | -6.40159235541700d-11 |
| -5.69997525494007d-11 | 3.46061668931529d-10 | 1.22296689879342d-11 |
| 2.00314796017624d-09 | -6.40159235541700d-11 | 9.94191359957619d-11 |
| 8.47200406101625d-11 | -1.51814214531290d-09 | -8.09744421672949d-10 |
| 1.67658416889601d-09 | -5.69997525494007d-11 | 8.47200406101625d-11 |
| 7.25020464385916d-11 | -1.20680507935355d-09 | -6.19868380959793d-10 |
| -3.93401117854579d-08 | 3.46061668931529d-10 | -1.51814214531290d-09 |
| -1.20680507935355d-09 | 5.93058700394746d-08 | 4.17661813028401d-08 |
| -2.33843803650305d-08 | 1.22296689879342d-11 | -8.09744421672949d-10 |
| -6.19868380959793d-10 | 4.17661813028401d-08 | 9.99999956738716d-08 |

Measurement noise covariance matrix $\theta_2$:

1.84412510991842d-02

Regulator gains G: (Also as computed in Chapter 2)

```
-7.64859358895862d-01   -3.38674675832647d-04   2.45537909052670d+00
-1.69155508507296d+00    1.04707683654752d+00   5.10491114597691d+00
```

Filter gains K:

```
 6.30001213506085d-03
-2.05415833543128d-01
 4.01197820069173d-03
 7.47232508540808d-03
-2.17948949924278d-03
-2.17948949924279d-03
```

The following tables present the data defining the 10 scaled structures analyzed in Chapters 5 and 6, and the optimized structure discussed in Chapter 8. Note that only the non-zero entries of the individual $\Psi$ matrices are shown. For all the structures the output node scaling parameter $\rho$ equals 0.02199717628337.

Structure (a)

Direct Form II
Number of Precedence Levels: 2
Number of Coefficients in Scaled Structure: 13
(non-zero, non-unity entries in the modified state space matrices)

| Non-zero entries in $\Psi_2$, $\Psi_1$ | | |
|---|---|---|
| Matrix | Index | Value |
| $\Psi_2$ | (7,1) | -2316.596730195619 |
| " | (7,2) | 17216.30907463747 |
| " | (7,3) | -46538.88776849179 |
| " | (7,4) | 60049.21454042759 |
| " | (7,5) | -37783.02361099942 |
| " | (7,6) | 9373.006832979322 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,3) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| $\Psi_1$ | (6,1) | -0.11903082227744 |
| " | (6,2) | 1.09870649812723 |
| " | (6,3) | -3.98894899287426 |
| " | (6,4) | 7.49594995996605 |
| " | (6,5) | -7.82430422984935 |
| " | (6,6) | 4.33762715269116 |
| " | (6,8) | 0.00010128626129 |
| " | (1,2) | 1.0 |
| " | (2,3) | 1.0 |
| " | (3,4) | 1.0 |
| " | (4,5) | 1.0 |
| " | (5,6) | 1.0 |

Structure (b)

Parallel Direct Form II, 4 first-order and 1 second-order sections
Number of Precedence Levels: 2
Number of Coefficients in Scaled Structure: 17
(non-zero, non-unity entries in the modified state space matrices)

Non-zero entries in $\Psi_2$, $\Psi_1$

| Matrix | Index | Value |
|---|---|---|
| $\Psi_2$ | (7,6) | 0.03890104412969 |
| " | (7,5) | 1.15283628631438 |
| " | (7,4) | 0.13875077275467 |
| " | (7,3) | -0.00460563493139 |
| " | (7,2) | 0.52228239125502 |
| " | (7,1) | -1.37949754700868 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,3) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| $\Psi_1$ | (2,2) | 1.46297047489118 |
| " | (2,1) | -0.69683507325690 |
| " | (6,8) | 0.87673782058497 |
| " | (3,3) | 0.99868711357757 |
| " | (5,8) | 0.64232806309622 |
| " | (4,4) | 0.99514095413908 |
| " | (4,8) | 0.17364017081712 |
| " | (5,5) | 0.58903698597208 |
| " | (3,8) | 0.15261498391194 |
| " | (6,6) | 0.29179162411121 |
| " | (2,8) | 0.28980851506818 |
| " | (1,2) | 1.0 |

Structure (c)

Parallel Direct Form II, 3 second-order sections
Number of Precedence Levels: 2
Number of Coefficients in Scaled Structure: 15
(non-zero, non-unity entries in the modified state space matrices)
Pole Pairing: (Refer to figure 5-7)

$z_{p1}$ and $z_{p4}$
$z_{p2}$ and $z_{p3}$
$z_{p5}$ and $z_{p6}$ (These are the complex poles)

Non-zero entries in $\Psi_2$, $\Psi_1$

| Matrix | Index | Value |
|---|---|---|
| $\Psi_2$ | (7,6) | 10.48075527883454 |
| " | (7,5) | -10.29571120349337 |
| " | (7,3) | -0.31185194361843 |
| " | (7,4) | 0.30767918685885 |
| " | (7,2) | 0.52228239125501 |
| " | (7,1) | -1.37949754700866 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,3) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| $\Psi_1$ | (2,2) | 1.46297047489119 |
| " | (2,1) | -0.69683507325690 |
| " | (4,3) | -0.29140853484973 |
| " | (4,4) | 1.29047873768878 |
| " | (6,5) | -0.58617482824346 |
| " | (6,6) | 1.58417794011116 |
| " | (6,8) | 0.07295197592120 |
| " | (4,8) | 0.10856479467707 |
| " | (2,8) | 0.28980851506819 |
| " | (1,2) | 1.0 |
| " | (3,4) | 1.0 |
| " | (5,6) | 1.0 |

Structure (d)

Parallel Direct Form II, 3 second-order sections
Number of Precedence Levels: 2
Number of Coefficients in Scaled Structures: 15
(non-zero, non-unity entries in the modified state space matrices)
Pole Pairing: (Refer to figure 5-7)

$z_{p1}$ and $z_{p2}$

$z_{p3}$ and $z_{p4}$

$z_{p5}$ and $z_{p6}$ (These are the complex poles)

Non-zero entries in $\Psi_2$, $\Psi_1$

| Matrix | Index | Value |
|---|---|---|
| $\Psi_2$ | (7,6) | 1.59834173340604 |
| " | (7,5) | -0.48730146270494 |
| " | (7,4) | 15.71737776482720 |
| " | (7,3) | -15.69841756881241 |
| " | (7,2) | 0.52228239125470 |
| " | (7,1) | -1.37949754700784 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,3) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| $\Psi_1$ | (2,2) | 1.46297047489118 |
| " | (2,1) | -0.69683507325690 |
| " | (4,3) | -0.99383444709201 |
| " | (4,4) | 1.99382806771667 |
| " | (6,5) | -0.17187605879836 |
| " | (6,6) | 0.88082861008328 |
| " | (6,8) | 0.48463047627064 |
| " | (4,8) | 0.00148815020744 |
| " | (2,8) | 0.28980851506825 |
| " | (1,2) | 1.0 |
| " | (3,4) | 1.0 |
| " | (5,6) | 1.0 |

Structure (e)

Parallel, One-level Version of (c)
Number of Precedence Levels: 1
Number of Coefficients in Scaled Structure: 16
(non-zero, non-unity entries in the modified state space matrices)
Pole Pairing: same as (c)

| Non-zero entries in $\Psi_1$ | | |
|---|---|---|
| Matrix | Index | Value |
| $\Psi_1^*$ | (2,1) | -0.696835073257 |
| " | (2,2) | 1.462970474891 |
| " | (2,8) | 0.289808515068 |
| " | (7,3) | -0.089660341046 |
| " | (4,3) | -0.291408534850 |
| " | (4,4) | 1.290478737689 |
| " | (4,8) | 0.108564794677 |
| " | (7,4) | 0.085201505052 |
| " | (6,5) | -0.586174828243 |
| " | (7,2) | -0.615413829047 |
| " | (6,6) | 1.584177940111 |
| " | (6,8) | 0.072951975921 |
| " | (7,1) | -0.363944688371 |
| " | (7,5) | -6.143554925433 |
| " | (7,6) | 6.307670104940 |
| " | (7,8) | 0.949356818741 |
| " | (1,2) | 1.0 |
| " | (3,4) | 1.0 |
| " | (5,6) | 1.0 |

Structure (f)

Block Optimal Parallel
Number of Precedence Levels: 1
Number of Coefficients in Scaled Structure: 25
(non-zero, non-unity entries in the modified state space matrices)
Pole Pairing: same as (c) and (e)

| Non-zero entries in $\Psi_1$ | | |
|---|---|---|
| Matrix | Index | Value |
| $\Psi_1$ | (2,1) | -0.33647827003132 |
| " | (2,2) | 0.68249200666952 |
| " | (2,8) | 0.65051552691033 |
| " | (7,3) | -0.08038901173235 |
| " | (4,3) | -0.20036428295682 |
| " | (4,4) | 1.19946355533120 |
| " | (4,8) | 0.11870639047793 |
| " | (7,4) | 0.07597348041937 |
| " | (6,5) | 0.19085044755223 |
| " | (7,2) | -0.43070856151277 |
| " | (6,6) | 0.73170685139682 |
| " | (6,8) | 0.45237970547959 |
| " | (7,1) | -0.73947570074840 |
| " | (7,5) | -0.54742490834586 |
| " | (7,6) | 0.94099544414975 |
| " | (7,8) | 0.94935681874100 |
| " | (1,1) | 0.78047846822148 |
| " | (1,2) | 0.48789111196657 |
| " | (3,3) | 0.09101518235780 |
| " | (3,4) | 0.90953905526798 |
| " | (5,5) | 0.85247108871418 |
| " | (5,6) | 0.19692962981701 |
| " | (1,8) | -0.13770626781352 |
| " | (3,8) | -0.00287783447672 |
| " | (5,8) | -0.06296709412667 |

Structure (g)

Cascade Direct Form II, 3 second-order sections
Number of Precedence Levels: 4
Number of Coefficients in Scaled Structure: 15
(non-zero, non-unity entries in the modified state space matrices)
Pole and Zero Pairing: (Refer to figure 5-7)

Section 1: $z_{p5}$ and $z_{p6}$, $z_{z1}$

Section 2: $z_{p3}$ and $z_{p4}$, $z_{z4}$ and $z_{z5}$

Section 3: $z_{p1}$ and $z_{p2}$, $z_{z2}$ and $z_{z3}$

| Non-zero entries in $\Psi_4$, $\Psi_3$, $\Psi_2$, $\Psi_1$ | | |
|---|---|---|
| **Matrix** | **Index** | **Value** |
| $\Psi_4$ | (7,6) | -1101.542292912427 |
| " | (7,5) | 541.2874849022007 |
| " | (7,7) | 560.2771331108011 |
| " | (1,2) | 1.0 |
| " | (2,1) | 1.0 |
| " | (3,4) | 1.0 |
| " | (4,3) | 1.0 |
| " | (5,6) | 1.0 |
| " | (6,7) | 1.0 |
| $\Psi_3$ | (7,6) | 0.88082861008329 |
| " | (7,5) | -0.17187605879836 |
| " | (7,4) | -6.00228882670692 |
| " | (7,3) | 2.89048675941179 |
| " | (7,7) | 3.40307257702863 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,7) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| $\Psi_2$ | (7,3) | 1.99382806771666 |
| " | (7,2) | -0.99383444709200 |
| " | (7,1) | -0.00051747678098 |
| " | (7,6) | 0.00171808332848 |
| " | (1,6) | 1.0 |
| " | (2,1) | 1.0 |
| " | (3,2) | 1.0 |
| " | (4,3) | 1.0 |
| " | (5,4) | 1.0 |
| " | (6,5) | 1.0 |
| $\Psi_1$ | (6,2) | 1.46297047489119 |
| " | (6,1) | -0.69683507325690 |
| " | (6,8) | 0.28980851498215 |
| " | (1,2) | 1.0 |
| " | (2,3) | 1.0 |
| " | (3,4) | 1.0 |
| " | (4,5) | 1.0 |
| " | (5,6) | 1.0 |

Structure (h)

Cascade Direct Form II, 3 second-order sections
Number of Precedence Levels: 4
Number of Coefficients in Scaled Structure: 15
(non-zero, non-unity entries in the modified state space matrices)
Pole and Zero Pairing: (Refer to figure 5-7)

Section 1: $z_{p2}$ and $z_{p3}$, $z_{z2}$

Section 2: $z_{p5}$ and $z_{p6}$, $z_{z4}$ and $z_{z5}$

Section 3: $z_{p1}$ and $z_{p4}$, $z_{z1}$ and $z_{z3}$

| Non-zero entries in $\Psi_4$, $\Psi_3$, $\Psi_2$, $\Psi_1$ | | |
|---|---|---|
| Matrix | Index | Value |
| $\Psi_4$ | (7,6) | -35.08378898367869 |
| " | (7,5) | 8.11873624443843 |
| " | (7,7) | 26.98802315299709 |
| " | (1,2) | 1.0 |
| " | (2,1) | 1.0 |
| " | (3,4) | 1.0 |
| " | (4,3) | 1.0 |
| " | (5,6) | 1.0 |
| " | (6,7) | 1.0 |
| $\Psi_3$ | (7,6) | 1.29047873768878 |
| " | (7,5) | -0.29140853484973 |
| " | (7,4) | -0.45738885908277 |
| " | (7,3) | 0.22026204990314 |
| " | (7,7) | 0.25932232325397 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,7) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| $\Psi_2$ | (7,3) | 1.46297047489118 |
| " | (7,2) | -0.69683507325690 |
| " | (7,1) | -1.79860505553554 |
| " | (7,6) | 1.85943686039663 |
| " | (1,6) | 1.0 |
| " | (2,1) | 1.0 |
| " | (3,2) | 1.0 |
| " | (4,3) | 1.0 |
| " | (5,4) | 1.0 |
| " | (6,5) | 1.0 |
| $\Psi_1$ | (6,2) | 1.58417794011116 |
| " | (6,1) | -0.58617482824346 |
| " | (6,8) | 0.07295197611457 |
| " | (1,2) | 1.0 |
| " | (2,3) | 1.0 |
| " | (3,4) | 1.0 |
| " | (4,5) | 1.0 |
| " | (5,6) | 1.0 |

Structure (i)

Cascade Direct Form I, 3 second-order sections
Number of Precedence Levels: 3
Number of Coefficients in Scaled Structure: 14
(non-zero, non-unity entries in the modified state space matrices)
Pole and Zero Pairing: same as (g)

Non-zero entries in $\Psi_3$, $\Psi_2$, $\Psi_1$

| Matrix | Index | Value |
|---|---|---|
| $\Psi_3$ | (7,3) | -320.6453445770277 |
| " | (7,2) | 157.5620956439277 |
| " | (7,5) | 0.88082861008329 |
| " | (7,4) | -0.17187605879836 |
| " | (7,8) | 163.0897474939010 |
| " | (1,6) | 1.0 |
| " | (2,1) | 1.0 |
| " | (3,7) | 1.0 |
| " | (4,3) | 1.0 |
| " | (5,8) | 1.0 |
| " | (6,5) | 1.0 |
| $\Psi_2$ | (8,2) | -0.02595431628362 |
| " | (8,1) | 0.01249866671420 |
| " | (8,4) | 1.99382806771669 |
| " | (8,3) | -0.99383444709202 |
| " | (8,8) | 0.01471512360540 |
| " | (1,2) | 1.0 |
| " | (2,3) | 1.0 |
| " | (3,4) | 1.0 |
| " | (4,5) | 1.0 |
| " | (5,6) | 1.0 |
| " | (6,7) | 1.0 |
| " | (7,8) | 1.0 |
| $\Psi_1$ | (8,1) | -0.11914766720671 |
| " | (8,8) | 0.39558416566143 |
| " | (8,3) | 1.46297047489118 |
| " | (8,2) | -0.69683507325690 |
| " | (1,2) | 1.0 |
| " | (2,3) | 1.0 |
| " | (3,4) | 1.0 |
| " | (4,5) | 1.0 |
| " | (5,6) | 1.0 |
| " | (6,7) | 1.0 |
| " | (7,8) | 1.0 |

Structure (j)

Simple
Number of Precedence Levels: 3
Number of Coefficients in Scaled Structure: 50
(non-zero, non-unity entries in the modified state space matrices)

| Non-zero entries in $\Psi_3$, $\Psi_2$, $\Psi_1$ | | |
|---|---|---|
| Matrix | Index | Value |
| $\Psi_3$ | (7,1) | 0.79382319292953 |
| " | (7,2) | 0.13324583104339 |
| " | (7,3) | -1.28133934418680 |
| " | (7,4) | 1.63323383955448 |
| " | (7,5) | -0.22354700928633 |
| " | (7,6) | -1.07427890614435 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,3) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| $\Psi_2$ | (1,1) | 0.89415889987584 |
| " | (1,2) | 0.02219872745941 |
| " | (1,3) | -0.40090758959435 |
| " | (1,4) | -0.00008683035986 |
| " | (1,5) | -0.17493645514225 |
| " | (1,6) | -0.12721034794726 |
| " | (2,1) | -0.00059340804849 |
| " | (2,2) | 0.99865886665303 |
| " | (2,3) | -0.00167440057433 |
| " | (2,4) | -0.00789688283429 |
| " | (2,5) | 0.00003836756094 |
| " | (2,6) | 0.00000711410910 |
| " | (3,1) | 0.18068685658836 |
| " | (3,2) | -0.00638575742111 |
| " | (3,3) | 0.84535736060977 |
| " | (3,4) | 0.00002869993983 |
| " | (3,5) | -0.02382581059578 |
| " | (3,6) | -0.01138138229375 |

| Matrix | Index | Value |
|---|---|---|
| $\Psi_2$ | (4,1) | 0.10382245521832 |
| " | (4,2) | 0.00119679543943 |
| " | (4,3) | -0.02270574399998 |
| " | (4,4) | 0.99999686644492 |
| " | (4,5) | -0.01171380959155 |
| " | (4,6) | -0.00509674027466 |
| " | (5,5) | 0.30119422754825 |
| " | (5,6) | 0.68880300468145 |
| " | (1,8) | 0.25341237237753 |
| " | (2,8) | -0.02208549707703 |
| " | (3,8) | 0.32520493273496 |
| " | (4,8) | 0.32736911273784 |
| " | (5,8) | -0.43182585076519 |
| " | (6,8) | -0.43809680729855 |
| " | (1,7) | -0.02223658684666 |
| " | (2,7) | 0.00000003108449 |
| " | (3,7) | -0.00155168069308 |
| " | (4,7) | -0.00064200329831 |
| " | (5,7) | 0.19562955072811 |
| " | (6,7) | 0.47519418802086 |
| " | (6,6) | 1.0 |
| $\Psi_1$ | (8,2) | -0.02874919859251 |
| " | (8,1) | -0.02486071250568 |
| " | (8,3) | -0.38589414084909 |
| " | (8,4) | -0.02282538209825 |
| " | (8,5) | -0.01812935994315 |
| " | (8,8) | 1.07470407782701 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,3) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| " | (7,7) | 1.0 |

**Optimized Structure Considered in Chapter 8**
**(Based on Structure (c))**

**Number of Precedence Levels: 2**
**Number of Coefficient in Scaled Structure: 17**
**(non-zero, non-unity entries in the modified state space matrices)**

| Non-zero entries in $\Psi_2$, $\Psi_1$ | | |
|---|---|---|
| Matrix | Index | Value |
| $\Psi_2$ | (7,6) | 1.34435168286127 |
| " | (7,5) | -0.50777452620114 |
| " | (7,3) | -0.31185194361846 |
| " | (7,4) | 0.30767918685888 |
| " | (7,2) | 0.52228239125501 |
| " | (7,1) | -1.37949754700866 |
| " | (1,1) | 1.0 |
| " | (2,2) | 1.0 |
| " | (3,3) | 1.0 |
| " | (4,4) | 1.0 |
| " | (5,5) | 1.0 |
| " | (6,6) | 1.0 |
| $\Psi_1$ | (2,2) | 1.46297047489118 |
| " | (2,1) | -0.69683507325689 |
| " | (4,3) | -0.29140853484973 |
| " | (4,4) | 1.29047873768877 |
| " | (6,5) | 0.16466105298259 |
| " | (6,6) | 0.65028182128291 |
| " | (6,8) | 0.56874389081747 |
| " | (4,8) | 0.10856479467706 |
| " | (2,8) | 0.28980851506819 |
| " | (5,5) | 0.93389611882824 |
| " | (5,6) | 0.12826858819766 |
| " | (1,2) | 1.0 |
| " | (3,4) | 1.0 |

## Appendix B: The Adjoint Lyapunov Operator

If we take the trace of the product of two matrices to be an inner product on the space of matrices, and $\pi$ to be a matrix operator, then:

$$\text{trace}\,(\pi(X)\ U) = \text{trace}\,(X\ \pi^*(U)) \tag{B.1}$$

where $\pi^*$ is the adjoint operator of $\pi$. For $\pi(X) = X - AXA'$, the operator $\pi^*$ can be derived from (B.1):

$$\begin{aligned}\text{trace}\,(\,(X-AXA')\,U\,) &= \text{trace}\,(XU) - \text{trace}\,(AXA'U)\\ &= \text{trace}\,(XU) - \text{trace}\,(XA'UA)\\ &= \text{trace}\,(X(U-A'UA)\,)\end{aligned} \tag{B.2}$$

Thus $\pi^*(U) = U - A'UA$.

As used in section 5.5, the Lyapunov equation (5.37) and the trace (5.38) were replaced by the equivalent equations (5.39) and (5.40). Relating this to the derivation above:

$$\begin{aligned} X &= V\\ A &= \Psi_{11}\\ U &= W_1\\ \pi^*(U) &= \Pi\\ \pi(X) &= \frac{\Delta_r^2}{12}\ \Lambda_1 S_1^{-2}\end{aligned} \tag{B.3}$$

## Appendix C: A Simplified Evaluation of (6.23)

In this appendix we will derive the expression used in the *SWL* and *MSWL* algorithms for computing the second partial derivatives of J. Evaluating this expression will be simpler than directly computing (6.22) and (6.23). Using (6.22) and the expressions in (6.25) and (6.26), and defining the following matrices:

$$D_1 = \left[\begin{array}{c|c|c} \Phi & 0_n & \Gamma k_{da} \\ \hline 0 & I_{n+1} & \\ \hline Lk_{ad} & 0 & \end{array}\right] \qquad \text{(C.1)}$$

$$D_2 = \begin{bmatrix} 0_n & 0 & 0 \\ 0 & 0_{n+1} & 0 \\ 0 & 0 & \theta_2 \end{bmatrix} \qquad \text{(C.2)}$$

we can rewrite $X_{ij}$ :

$$X_{ij} = \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{\partial \Psi_\infty}{\partial c_j} \end{bmatrix} D_1 \frac{\partial Z}{\partial c_i} A' + \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{\partial \Psi_\infty}{\partial c_i} \end{bmatrix} D_1 \frac{\partial Z}{\partial c_j} A'$$

$$+ \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{\partial \Psi_\infty}{\partial c_i} \end{bmatrix} \left( D_1 Z D_1' + D_2 \right) \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{\partial \Psi_\infty'}{\partial c_j} \end{bmatrix}$$

$$+\begin{bmatrix} 0 & 0 \\ 0 & \dfrac{\partial^2 \Psi_\infty}{\partial c_i \partial c_j} \end{bmatrix} \left( D_1 Z D_1' + D_2 \right) \begin{bmatrix} I_n & 0 \\ 0 & \Psi_\infty' \end{bmatrix} \tag{C.3}$$

Thus $X_{ij}$ will be a matrix whose lower right-hand $(n+1)\times(n+2)$ portion is non-zero, and the rest zero. Thus the trace expression in (6.23) can be simplified:

$$\frac{\partial^2 J}{\partial c_i \partial c_j} = 2 \text{ trace} \left\{ \frac{\partial \Psi_\infty}{\partial c_j}(M1)\frac{\partial Z}{\partial c_i}(M2) + \frac{\partial \Psi_\infty}{\partial c_i}(M1)\frac{\partial Z}{\partial c_j}(M2) \right\}$$

$$+ 2 \text{ trace} \left\{ \frac{\partial \Psi_\infty}{\partial c_i}(M3)\frac{\partial \Psi_\infty}{\partial c_j}(M4) + \frac{\partial^2 \Psi_\infty}{\partial c_i \partial c_j}(M5) \right\} \tag{C.4}$$

where $M1$, $M2$, $M3$, and $M4$ are precomputed matrices (computed only once for all $i$ and $j$) the fixed matrices $D_1$, $Z$, $A$, $D_2$, $\hat{U}$, and $\Psi_\infty$. As it is shown in (C.4), a maximum three matrix multiplications and a trace operation are required for each term in (C.4), *for each i and j*. Thus in terms of operation counts, the calculation of (6.23) would be roughly proportional to $(N^2)(2n+1)^3$.

In fact, this expression can be further simplified to reduce the computational load. By substituting (6.27) and (6.28) into the partial derivatives $\dfrac{\partial \Psi_\infty}{\partial c_i}$, $\dfrac{\partial \Psi_\infty}{\partial c_j}$, and $\dfrac{\partial^2 \Psi_\infty}{\partial c_i \partial c_j}$, applying simple trace identities, and combining the matrices $\Psi_1$, $\Psi_2$, $\Psi_3 \cdots \Psi_q$ with $M1$, $M2$, $M3$, $M4$, and $M5$, we can produce:

$$\frac{\partial^2 J}{\partial c_i \partial_j} = 2 \text{ trace } \left\{ (M6) E_{kl} (M7) \frac{\partial Z}{\partial c_j} \right\}$$

$$+ 2 \text{ trace } \left\{ (M8) E_{rs} (M9) \frac{\partial Z}{\partial c_i} \right\}$$

$$+ 2 \text{ trace } \left\{ E_{kl} (M10) E_{rs} (M11) \right\}$$

$$+ 2 \text{ trace } \left\{ E_{ks} (M12) \right\} \quad \text{if} \quad l = r \qquad \text{(C.5)}$$

where the precomputable matrices $M7$, $M8$, $M9$, $M10$, $M11$, and $M12$ will depend on which specific precedence-level matrices contain coefficients $c_i$ and $c_j$. As the number of precedence levels goes up, so does the number of such matrices — but they can still all be precomputed. Equation (C.5) can be simplified by taking advantage of the special form of $E_{kl}$ and $E_{rs}$ (described in section 6.4). For the first trace term of (C.5), we can write:

$$(M6)\, E_{kl}\, (M7) = (V1)\, (V2) \qquad \text{(C.6)}$$

where $V1$ is a column $(2n+1)$-vector equal to the $k^{th}$ column of $M6$ and $V2$ is a row $(2n+1)$-vector equal to the $l^{th}$ row of $M7$. Thus the first term of (C.5) can be written as:

$$2 \text{ trace } (V1)\, (V2) \frac{\partial Z}{\partial c_j} = 2 \text{ trace } (V2) \frac{\partial Z}{\partial c_j} (V1) = 2\, (V2) \frac{\partial Z}{\partial c_j} (V1) \qquad \text{(C.7)}$$

Now, only one vector-matrix multiplication and one vector dot product are required per $i$ and $j$. In terms of operation counts, this simplification reduces the calcula-

tion of (6.23) (given the first partial derivatives of $Z$) from being roughly proportional to $N^2(2n+1)^3$ to being proportional to $N^2(2n+1)^2$, a large savings.

The second term of (C.5) can be simplified in exactly the same manner as term 1. The third term, since there is no dependence on $c_l$ or $c_j$ other than in $E_{kl}$ and $E_{rs}$, we can reduce to:

$$2\,\text{trace}\left\{E_{kl}(M10)E_{rs}(M11)\right\} = 2\,M10(l,r)M11(s,k) \qquad \text{(C.8)}$$

This involves even less computation then the first two terms. Finally, the fourth term reduces to the simplest form of all:

$$2\,\text{trace}\left\{E_{ks}(M12)\right\} = 2\,M12(s,k) \qquad \text{(C.9)}$$

Thus overall, the number of operations involved in computing this simplified expression will be proportional to $N^2(2n+1)^2$ where $N$ is the number of rounded coefficients in the structure, and $n$ is the plant order.

## REFERENCES


[1] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, J. Wiley & Sons, New York, New York, 1972.

[2] J.C. Willems and S.K. Mitter, "Controllability, Observability, Pole Allocation, and State Reconstruction," *IEEE Trans. on Aut. Control*, Vol. AC-16, No. 6, December 1971, pp. 582-595.

[3] M. Athans, "The Discrete Time Linear-Quadratic-Gaussian Stochastic Control Problem," *Annals of Economic and Social Measurement*, Vol. 1, No. 4, 1972, pp. 449-491.

[4] B.C. Kuo, *Analysis and Synthesis of Sampled-Data Control Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

[5] D.M. Auslander, Y. Takahashi, and M. Tomizuka, "Direct Digital Process Control: Practice and Algorithms for Microprocessor Application," *Proc. IEEE*, Vol. 66, No. 2, February 1978, pp. 199-208.

[6] T.F. Tao, D. Bar Yehoshua, and R. Martinez, "Applications of Microprocessors in Control Problems," *Proc. 1977 Joint Automatic Control Conf.*, 1977, pp. 8-13.

[7] J.B. Knowles and R. Edwards, "Effect of a Finite-Word-Length Computer in a Sampled-Data Feedback System," *Proc. IEE*, Vol. 112, No. 6, June 1965, pp. 1197-1207.

[8] E.E. Curry, "The Analysis of Round-Off and Truncation Errors in a Hybrid Control System," *IEEE Trans. on Aut. Control*, Vol. AC-13, October 1967, pp. 601-604.

[9] J.E. Bertram, "The Effect of Quantization in Sampled-feedback Systems," *Trans. Amer. Inst. Elec. Engrs.*, Vol. 77, Pt. 2, September 1958, pp. 177-182.

[10] J.B. Slaughter, "Quantization Errors in Digital Control Systems," *IEEE Trans. Aut. Control*, Vol. AC-9, No. 1, January 1964, pp. 70-74.

[11] G.W. Johnson, "Upper Bound on Dynamic Quantization Error in Digital Control Systems via the Direct Method of Lyapunov," *IEEE Trans. Aut. Control*, Vol. AC-10, No. 4, October 1965, pp. 439-448.

[12] G.N.T. Lack and G.W. Johnson, "Comments on "Upper Bound On Dynamic Quantization Error in Digital Control Systems Via the Direct Method of Liapunov"," *IEEE Trans. Aut. Control*, Vol. AC-11, April 1966, pp. 331-334.

[13] A.B. Sripad, "Models for Finite Precision Arithmetic, With Application to the Digital Implementation of Kalman Filters," Sc. D. Dissertation, Washington Univ., Sever Institute, January 1978.

[14] R.E. Rink and H.Y. Chong, "Performance of State Regulator Systems with Floating-Point Computation," *IEEE Trans. on Aut. Control*, Vol. AC-24, No. 3, June 1979, pp. 411-421.

[15] F.A. Farrar, "Microprocessor Implementation of Advanced Control Modes," *Summer Computer Simulation Conference Proceedings*, Chicago, Illinois, July 1977, pp. 339-342.

[16] A.S. Willsky, "Digital Signal Processing and Control and Estimation Theory — Points of Tangency, Areas of Intersection, and Parallel Directions," MIT ESL Rept. ESL-R-712, Cambridge, Mass., January 1977.

[17] D.S.K. Chan, "Theory and Implementation of Multidimensional Discrete Systems for Signal Processing," Ph.D. Dissertation, MIT, Department of Electrical Engineering and Computer Science, May 1978.

[18] C.T. Mullis and R.A. Roberts, "Synthesis of Minimum Roundoff Noise Fixed-Point Digital Filters," *IEEE Trans. Circuits & Systems*, Vol. CAS-23, No. 9, September 1976, pp. 551-562.

[19] W.S. Levine, T.L. Johnson, and M. Athans, "Optimal Limited State Variable Feedback Controllers for Linear Systems," *IEEE Trans. Aut. Control*, Vol. AC-16, No. 6, December 1971, pp. 785-893.

[20] D.P. Looze, P.K. Houpt, N.R. Sandell,Jr., and M. Athans, "On Decentralized Estimation and Control with Application to Freeway Ramp Metering," *IEEE Trans. on Aut. Control*, Vol. AC-23, No. 2, April 1978, pp. 268-275.

[21] J.F. Kaiser, "On the Limit Cycle Problem," *Proc. IEEE Inter. Conf. Acous. Speech & Signal Processing*, 1976, pp. 642-644.

[22] M. Athans, Guest Ed., *IEEE Transactions Aut. Control, Special Issue on Linear-Quadratic-Gaussian Problem*, Vol. AC-16, No. 6, December 1971.

[23] M.G. Safonov and M. Athans, "Gain and Phase Margin for Multiloop LQG Regulators," *Proc. IEEE Trans. on Aut. Control*, Vol. AC-22, No. 2, April 1977, pp. 173-179.

[24] C.A. Harvey and G. Stein, "Quadratic Weights for Asymptotic Regulator Properties," *Proc. 1977 IEEE Conf. Decision & Control*, Vol. 1, 1977, pp. 1220-1228.

[25] G.K. Roberts, "Consideration of Computer Limitations in Implementing On-Line Controls," MIT ESL Rept. ESL-R-665, Cambridge, Mass. June 1976.

[26] A. Gelb, ed., *Applied Optimal Estimation*, MIT Press, Cambridge, Mass., 1974.

[27] A.P. Sage, *Optimal Systems Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1968.

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

[29] L.R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1975.

[30] S.A. Tretter, *Introduction to Discrete-Time Signal Processing*, J. Wiley & Sons, New York, New York, 1976.

[31] R.E. Crochiere and A.V. Oppenheim, "Analysis of Linear Digital Networks," *Proc. IEEE*, Vol. 63, No. 4, April 1975, pp. 581-595.

[32] R.E. Crochiere, "Digital Network Theory and its Application to the Analysis and Design of Digital Filters," Ph.D. Dissertation, MIT, Department of Electrical Engineering, April 1974.

[33] J.F. Kaiser, "Some Practical Considerations in the Realization of Linear Digital Filters," *Proc. Third Annual Allerton Conf. Circuit & System Theory*, October 1965, Monticello, Illinois, pp. 621-633.

[34] E. Avenhaus, "A Proposal to Find Suitable Canonic Structures for the Implementation of Digital Filters with Small Coefficient Wordlength," *Nachr. Tech. Zeit.*, Vol. 25, No. 8, August 1972, pp. 377-382.

[35] E. Lueder and K. Haug, "Calculation of All Equivalent and Canonic 2nd Order Digital Filter Structures," *Proc. 1978 IEEE Inter. Conf. Acous. Speech & Signal Processing*, Tulsa, Oklahoma, April 1978, pp. 51-54.

[36] R.W. Brockett, *Finite Dimensional Linear Systems*, J. Wiley and Sons, Inc., New York, New York, 1970.

[37] C.T. Mullis and R.A. Roberts, "Roundoff Noise in Digital Filters — Frequency Transformations and Invariants," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-24, No. 6, December 1976, pp.538-550.

[38] C.T. Mullis and R.A. Roberts, "Filter Structures Which Minimize Roundoff Noise in Fixed-Point Digital Filters," *Proc. IEEE Inter. Conf. Acous. Speech & Signal Processing*, April 1976, Philadelphia, Penn., pp. 505-508.

[39] S.Y. Hwang, "Minimum Uncorrelated Unit Noise in State-Space Digital Filters," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-25, No. 4, August 1977, pp. 273-281.

[40] L.B. Jackson, A.G. Lindgren, and Y. Kim, "Synthesis of State-Space Digital Filters with Low Roundoff Noise and Coefficient Sensitivity," *Proc. IEEE Inter. Symp. Circuits & Systems*, April 1977, Phoenix, Arizona, pp. 41-44. (also *IEEE Trans. Circuits & Systems*, Vol. CAS-26, No. 3, March 1979, pp. 149-153.)

[41] A.H. Gray, Jr. and J.D. Markel, "Digital Ladder and Lattice Filter Synthesis," *IEEE Trans. Audio & Electroacoustics*, Vol. AU-21, No. 6, December 1974, pp. 491-500.

[42] S.K. Mitra and R.J. Sherwood, "Canonic Realizations of Digital Filters Using the Continued Fraction Expansion," *IEEE Trans. Audio & Electroacoustics*, Vol. AU-20, No. 3, August 1972, pp. 185-194.

[43] A. Fettweis, "Digital Filter Structures Related to Classical Filter Networks", *Arch. Elek. Übertr.*, Vol. 25, No. 2, February 1971, pp. 79-89.

[44] R.E. Crochiere, "Digital Ladder Structures and Coefficient Sensitivity," *IEEE Trans. Audio & Electroacoustics*, Vol. AU-20, No. 4, October 1972, pp. 240-246.

[45] K. Renner and S.C. Gupta, "On the Design of Wave Digital Filters With Low Sensitivity Properties," *IEEE Trans. Circuit Theory*, Vol. CT-20, No. 5, September 1973, pp. 555-567.

[46] A. Fettweis, "Pseudopassivity, Sensitivity, and Stability of Wave Digital Filters," *IEEE Trans. Circuit Theory*, Vol. CT-19, No. 6, November 1972, pp. 668-673.

[47] A. Fettweis and K. Meerkötter, "Suppression of Parasitic Oscillations in Wave Digital Filters," *IEEE Trans. Circuits & Systems*, Vol. CAS-22, March 1975, pp. 239-246, and June 1975, p. 575.

[48] A. Fettweis, "On Sensitivity and Roundoff Noise in Wave Digital Filters," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-22, No. 5, October 1974, pp. 383-384.

[49] A. Sedlmeyer and A. Fettweis, "Digital Filters With True Ladder Configuration," *Inter. Jour. of Circuit Theory and Applic.*, Vol. 1, No. 1, March 1973, pp. 5-10.

[50] A. Fettweis, "Wave Digital Filters with Reduced Number of Delays," *Inter. Jour. of Circuit Theory and Applic.*, Vol. 2, No. 4, 1974, pp. 319-320.

[51] K. Meerkötter and W. Wegener, "A New Second-Order Digital Filter Without Parasitic Oscillations," *Arch. Elek. Übert.*, Vol. 29, No. 7/8, July/August 1975, pp. 312-314.

[52] J. Allen and R.G. Gallagher, *Computation Structures*, MIT Course Notes for 6.032, 1977.

[53] S.K. Tewksbury, R.B. Kieburtz, J.S. Thompson, and S.P. Verma, "Tutorials on Signal Processing for Communications: Part II — Digital Signal Processing Architecture," *IEEE Communications Society Magazine*, January 1978, pp. 23-27.

[54] J. Allen, "Computer Architecture for Signal Processors," *Proc. IEEE*, Vol. 63, No. 4, April 1975, pp. 624-633.

[55] L.B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," *Bell Syst. Tech. Journal*, Vol. 49, No. 2, February 1970, pp. 159-184.

[56] S.Y. Hwang, "Dynamic Range Constraints in State-Space Digital Filtering," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-23, No. 6, December 1975, pp. 591-593.

[57] A.V. Oppenheim and C.J. Weinstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," *Proc. IEEE*, Vol. 60, August 1972, pp. 957-976.

[58] A.B. Sripad and D.L. Snyder, "Necessary and Sufficient Conditions for Quantization Errors to be Uniform & White," *IEEE Trans. Acous. Speech & Signal Process*, Vol. ASSP-25, No. 5, October 1977, pp. 442-448.

[59] T.A.C.M. Claasen, W.F.G. Mecklenbräuker, and J.B.H. Peek, "Quantization Noise Analysis for Fixed-Point Digital Filters Using Magnitude Truncation," *IEEE Trans. Circuit & Systems*, Vol. CAS-22, No. 11, Nov. 1975, pp. 887-895.

[60] T.A.C.M. Claasen, W.F.G. Mecklenbräuker, and J.B.H. Peek, "Effects of Quantization and Overflow in Recursive Digital Filters," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-24, No. 6, December 1976, pp. 517-529.

[61] L.B. Jackson, "Roundoff Noise Analysis for Fixed-Point Digital Filters Realized in Cascade or Parallel Form," *IEEE Trans. Audio & Electroacoustics*, Vol. AU-18, June 1970, pp. 107-122.

[62] S.K. Mitra, K. Hirano, and H. Sakaguchi, "A Simple Method of Computing the Input Quantization and Multiplication Roundoff Errors in a Digital Filter," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-22, No. 5, October 1974, pp. 326-329.

[63] S.Y. Hwang, "Roundoff Noise in State-Space Digital Filtering: A General Analysis," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-24, No. 3, June 1976, pp. 256-262.

[64] A.E. Bryson, Jr., Guest Ed. Mini-Issue on the F-8 DFBW, *IEEE Trans. Aut. Control*, Vol. AC-22, No. 5, October 1977, pp. 752-806.

[65] G. Dehner, "A Contribution to the Optimization of Roundoff-Noise in Recursive Digital Filters," *Arch. Elek. Über.*, Vol. 29, No. 12, December 1975, pp. 505-510.

[66] S.Y. Hwang, "On Optimization of Cascade Fixed-Point Digital Filters," *IEEE Trans. Circuit & Systems*, Vol. CAS-21, No. 1, January 1974, pp. 163-165.

[67] E. Avenhaus, "On the Design of Digital Filters with Coefficients of Limited Word length," *IEEE Trans. Audio & Electroacoustics*, Vol. AU-20, August 1972, pp. 206-212.

[68] K. Steiglitz, "Design Short-word Recursive Digital Filters," *Proc. 9th Allerton Conf.*, October 1971, pp. 778-788.

[69] N.I. Smith, "A Random-Search Method for Designing Finite-Wordlength Recursive Digital Filters," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-27, No. 1, February 1979, pp. 40-46.

[70] C.M. Rader and B. Gold, "Effects of Parameter Quantization on the Poles of a Digital Filter," *Proc. IEEE*, Vol. 55, May 1967, pp. 688-689.

[71] A.I. Abu-El-Haija, K. Shenoi, and A.M. Peterson, "A Structure Suitable for Implementing Digital Filters with Poles Near +1," *National Telecommunication Conf.*, 1977, pp. 29:5-1 - 29:5-8.

[72] R.C. Agarwal and C.S. Burrus, "New Recursive Digital Filter Structures Having Low Sensitivity and Roundoff Noise," *IEEE Trans. Circuit & Systems*, Vol. CAS-22, No. 12, December 1975, pp. 921-927.

[73] J.B. Knowles and E.M. Olcayto, "Coefficient Accuracy and Digital Filter Response," *IEEE Trans. Circuits & Systems*, Vol. CAS-15, March 1968, pp. 31-41.

[74] R.E. Crochiere, "A New Statistical Approach to the Coefficient Word Length Problem for Digital Filters," *IEEE Trans. Circuits & Systems*, Vol. CAS-22, No. 3, March 1975, pp. 190-196.

[75] D.S.K. Chan and L.R. Rabiner, "Analysis of Quantization Errors in the Direct Form for Finite Impulse Response Digital Filters," *IEEE Trans. Audio Electroacoustics*, Vol. AU-21, August 1973, pp. 354-366.

[76] A.Y. Barraud, "A Numerical Algorithm to Solve $A^TXA-X=Q$," *IEEE Trans. Aut. Control*, Vol. AC-22, No. 5, October 1977, pp. 883-885.

[77] A. Fettweis, "Roundoff Noise and Attenuation Sensitivity in Digital Filters with Fixed-Point Arithmetic," *IEEE Trans. Circuit Theory*, Vol. CT-20, No. 2, March 1973, pp. 174-175.

[78] A. Fettweis, "On the Connection Between Multiplier Wordlength Limitations and Roundoff Noise in Digital Filters," *IEEE Trans. Circuit Theory*, Vol. CT-19, No. 5, September 1972, pp. 486-491.

[79] R.B. Kieburtz, "An Experimental Study of Roundoff Effects in a 10th-Order Recursive Digital Filter," *IEEE Trans. Communications*, Vol. COM-21, No. 6, June 1973, pp. 757-763.

[80] T.A.C.M. Claasen, W.F.G. Mecklenbräuker, and J.B.H. Peek, "Frequency-Domain Criteria for the Absence of Zero-Input Limit Cycles in Nonlinear Discrete-Time Systems, With Applications to Digital Filters," *IEEE Trans. Circuit & Systems*, Vol. CAS-22, No. 3, March 1975, pp. 232-239.

[81] D.D. Siljak, "Algebraic Criteria for Positive Realness Relative to the Unit Circle," *J. Franklin Inst.*, Vol. 296, No. 2, August 1973, pp. 115-122.

[82] W.L. Mills, C.T. Mullis, and R.A. Roberts, "Digital Filters Without Overflow Oscillations, *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-26, No. 4, August 1978, pp. 334-338.

[83] C.W. Barnes and A.T. Fam, "Minimum Norm Recursive Digital Filters that Are Free of Overflow Limit Cycles," *IEEE Trans. Circuits & Systems*, Vol. CAS-24, No. 10, October 1977, pp. 569-574.

[84] C.W. Barnes, "Roundoff Noise and Overflow in Normal Digital Filters," *IEEE Trans. Circuits & Systems*, Vol. CAS-26, No. 3, March 1979, pp. 154-159.

[85] A.T. Fam and C.W. Barnes, "Nonminimal Realizations of Fixed-Point Digital Filters That Are Free of All Finite Word-Length Limit Cycles," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-27, No. 2, April 1979, pp. 149-153.

[86] L.B. Jackson, "Limit Cycles in State-Space Structures for Digital Filters," *IEEE Trans. Circuits & Systems*, Vol. CAS-26, No. 1, January 1979, pp. 67-68.

[87] S.R. Parker and S.F. Hess, "Limit Cycle Oscillations in Digital Filters," *IEEE Trans. Circuit Theory*, Vol. CT-18, No. 6, November 1971, pp. 687-697.

[88] L.B. Jackson, "An Analysis of Limit Cycles Due to Multiplicative Rounding in Recursive Digital Filters," *Proc. 7th Allerton Conf. Circuit & System Theory*, Monticello, Illinois, October 1969, pp. 69-78.

[89] I.W. Sandberg and J.F. Kaiser, "A Bound on Limit Cycles in Fixed-Point Implementations of Digital Filters," *IEEE Trans. Audio & Electroacoustics*, Vol AU-20, No. 2, June 1972, pp. 110-112.

[90] R.B. Kieburtz, V.B. Lawrence, and K.V. Mina, "Control of Limit Cycles in Recursive Digital Filters By Randomized Quantization," *IEEE Trans. Circuits & Systems*, Vol. CAS-24, No. 6, June 1977, pp. 291-299.

[91] G. Zames and N.A. Shneydor, "Dither in Nonlinear Systems," *IEEE Trans. Aut. Control*, Vol. AC-21, No. 5, October 1976, pp. 660-667.

[92] G. Zames and N.A. Shneydor, "Structural Stabilization and Quenching by Dither in Nonlinear Systems," *IEEE Trans. Aut. Control*, Vol. AC-22, No. 3, June 1977, pp. 352-361.

[93] V.B. Lawrence and K.V. Mina, "Control of Limit Cycle Oscillations in Second-Order Recursive Digital Filters Using Constrained Random Quantization," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-26, No. 2, April 1978, pp. 127-134.

[94] M. Büttner, "Elimination of Limit Cycles in Digital Filters with Very Low Increase in Quantization Noise," *IEEE Trans. Circuits & Systems*, Vol. CAS-24, No. 6, June 1977, pp. 300-304.

[95] A.N. Willson, Jr., "Limit Cycles Due to Adder Overflow in Digital Filters," *IEEE Trans. Circuit Theory*, Vol. CT-19, No. 4, July 1972, pp. 342-346.

[96] P.M. Ebert, J.E. Mazo, and M.G. Taylor, "Overflow Oscillations in Digital Filters," *Bell Sys. Tech. Journal*, Vol. 48, No. 9, Nov. 1969, pp. 2999-3020.

[97] T.A.C.M. Claasen, W.F.G. Mecklenbräuker, and J.B.H. Peek, "On the Stability of the Forced Response of Digital Filters with Overflow Nonlinearities," *IEEE Trans. Circuits & Systems*, Vol. 22, No. 8, August 1975, pp. 692-696.

[98] T.A.C.M. Claasen and L. Kristiansson, "Necessary and Sufficient Conditions for the Absence of Overflow Oscillations in 2nd-Order Recursive Digital Filters," *IEEE Trans. Acous. Speech & Signal Processing*, Vol. ASSP-23, No. 6, December 1975, pp. 509-515.

[99] A.N. Willson, Jr., "Some Effects of Quantization and Adder Overflow on the Forced Response of Digital Filters," *Bell Syst. Tech. Journal*, Vol. 51, No. 4, April 1972, pp. 863-887.

[100] A. Fettweis and K. Meerkötter, "On Parasitic Oscillation in Digital Filters Under Looped Conditions," *IEEE Trans. Circuit and Systems*, Vol. CAS-24, No. 9, September 1977, pp. 475-481.

[101] D.S.K. Chan, "Constrained Minimization of Roundoff Noise in Fixed-Point Digital Filters," *Proc. 1979 IEEE Inter. Conf. Acous. Speech & Signal Processing*, Washington, D.C., April 1979, pp. 335-339.

[102] S.M. Selby, ed., *CRC Standard Math Tables*, 19th edition, The Chemical Rubber Co., Cleveland, Ohio, 1971.

[103] S.R. Parker and P.E. Girard, "Correlated Noise Due to Roundoff in Fixed Point Digital Filters," *IEEE Trans. Circuits and Systems*, Vol. CAS-23, No. 4, April 1976, pp. 204-211.

[104] A.D. Edgar and S.C. Lee, "FOCUS Microcomputer Number System," *Communications of the ACM*, Vol. 22, No. 3, March 1979, pp. 166-177.