(65)

Finite Wordlength Effects in Fixed-Point Implementations of Linear Systems

by

Vinay Mohta

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1998

Author ....
Department of Electrical Engineering and Computer Science
May 22, 1998

Certified by..
George C. Verghese
Professor
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Finite Wordlength Effects in Fixed-Point Implementations of Linear Systems

by

Vinay Mohta

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 1998, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Finite wordlength effects in digital filters and controllers have been explored over the last three decades. Much of the work has originated from the digital signal processing community, with research activity in digital controllers increasing only recently as they have become more widespread. This thesis comprehensively surveys much of the research in both these areas. It specifically focuses on the works that deal with the fixed-point two's complement numeric representation and on those works that describe performance measures and the optimizations which minimize some of these measures.

The thesis ambitiously attempts to paint a cohesive picture of the research, utilizing the current filter and controller design process as the canvas. The hope is to stimulate significant new research with the questions that surface a consequence of such a cross-fertilization as well as to introduce and acquaint the reader with the primary tools and techniques of the FWL research community. A short introduction to the fixed-point numeric representation is provided.

Thesis Supervisor: George C. Verghese
Title: Professor

Acknowledgments

I would like to thank many people for many different things. This thesis has been the culmination of a lot of work, not just in the past year of graduate work, but also throughout my earlier four years here at MIT as an undergraduate. Many of my teachers inspired me and kept the spirit of mathematics and engineering alive, even on those painfully sleep deprived nights.

Specifically, I'd like to thank my thesis advisor, Professor George Verghese, who helped me select work and supported me at all the times that I needed it. I'd also like to thank Andy Bartlett of The MathWorks, Inc. for originally suggesting finite wordlength effects as a research area and posing the question that ultimately lead to this thesis, "How do I optimally implement this digital filter or controller?"

To the thesis gang (Hussein, Lawrence, Michael, Rob, and Sami), it's finally over! See ya on the other side! To all my friends, you know who you are, thanks for helping me keep my sanity and for teaching me so many other things that I never thought one year of graduate school possibly could. What a long, strange trip it's been...

A very special thanks to my brother, Vivek, for helping me out with all of my math questions whenever I was stuck and for all his love and support, and of course, most of all to my parents, for everything from sending me to this great institution to standing with me through the good and the bad.

My life is irrevocably altered!!

Mientras hay vida, hay esperanza.

Contents

1 Introduction **10**

2 Background **17**

2.1 The Fixed-Point Representation 17

2.2 FWL Effects and LTI Systems 22

2.3 Scaling 25

2.4 Summary 27

3 Controls and Signal Processing **29**

3.1 The Design Process: A Summary 30

3.2 The Design Process: A Bigger Picture 34

3.3 Implementation Details 41

4 Performance Metrics **43**

4.1 The $M_{L_{12}}$ Measure 45

4.2 The M_{L_2} Sensitivity Measure 53

4.3 Pole-Zero Sensitivity 57

4.3.1 Pole Sensitivity Minimization 60

4.3.2 Zero Sensitivity Minimization 63

4.3.3 Combined Pole-Zero Sensitivity Minimization 65

4.4 Frequency Weighted Sensitivity Minimization 66

4.5 Roundoff Noise Gain 68

4.6 Stochastic Models of Coefficient Quantization 72

4.7 A Synthetic Measure 74

4.8 The Controls Context 77

4.9 Closed-Loop Sensitivity 81

4.10 Stability Robustness Measures 84

4.11 Closed-Loop Noise Gain 89

4.12 Other Measures 91

5 Optimizations **92**

5.1 Realization Structures 93

5.1.1 Direct Forms I and II 94

5.1.2 Cascade Decomposition 97

5.1.3 Parallel Decomposition 100

5.1.4 Other Filter Structures 101

5.1.5 Non-Minimal Realizations 102

5.2 Minimum $M_{L_{12}}$ Sensitivity Realization 103

5.3 Minimum M_{L_2} Sensitivity Realization 105

5.4 Minimum M_{pz} Sensitivity Realization 107

5.5 Minimum Frequency-Weighted $M_{L_{12}}$ Realizations 109

5.6 Minimum Roundoff Noise Structures 110

5.6.1 Sectional Optimal Structures 115

5.6.2 Block-Optimal Structures 116

5.6.3 Non-Minimal Realizations 116

5.7 Minimum Synthetic Noise Gain Realizations 117

5.8 Block Processing Filters 117

5.9 Discrete (or Digital) Redesign 121

5.9.1 Bilinear Transform 122

5.9.2 Controller Discretization 123

5.9.3 Fast discretization of the plant 124

5.10 Minimum LQG Cost Realization 128

5.11 Sparseness Considerations 134

5.12 Other Optimizations 140

6 Conclusion **146**

A Notation **150**

B Definitions **151**

C Derivations **162**

C.1 Hadamard's Inequality 162

C.2 Hölder's Inequality 163

C.3 Transfer Function Pole Sensitivity Analysis 164

C.4 Matrix Derivatives 169

C.5 Using the Cauchy-Schwarz Inequality to bound $M_{L_{12}}$ 171

List of Figures

2-1 Fixed-point quantization characteristics. 18

2-1(a) Quantization plot using roundoff 18

2-1(b) Quantization plot using truncation 18

2-2 Overflow characteristics. 19

2-2(a) Saturation characteristic 19

2-2(b) Wrapping characteristic 19

2-3 Coefficient quantization modeled as an additive perturbation. 24

2-4 Scaling implementation. 27

3-1 FWL DT filter design. 30

3-2 A summary of the FWL design process. 30

3-3 An example of intersample ripple. 32

3-4 The digital filter/controller design process. 35

3-5 FWL DT filter error. 36

3-6 FWL DT filter error. 36

3-7 FWL DT controller error. 38

3-8 FWL DT controller error. 39

3-9 FWL DT controller error. 40

4-1 Multiplier with roundoff error modeled as noise. 68

4-2 Feedback-based state estimate observer. 78

4-3 A hybrid system. 82

5-1 Direct Form II. 95

5-2 Direct Form I. 96

5-3 Direct Form II Transpose. 97

5-4 Direct Form I Transpose. 142

5-5 Cascade decomposition of $H(z)$. 143

5-6 Parallel decomposition of $H(z)$. 143

5-7 FWL filter error. 143

5-8 FWL DT filter error. 143

5-9 Continuous time system. 144

5-10 Digital redesign. 144

5-11 A lifted closed-loop system. 144

5-12 DT controller approximation error. 144

5-13 Approximation error as a disturbance. 145

B-1 Arithmetic Components 151

B-1(a) Add Element 151

B-1(b) Multiply Element 151

B-1(c) Delay Element 151

B-2 A Hybrid System 155

List of Tables

2.1 Common numeric representations. 20

2.2 Range of internal variables $v[k] = (f*u)[k]$ for various classes of inputs u. 26

4.1 Measures. 46

Chapter 1

Introduction

Either poets must become engineers, or engineers must become poets.
-Norbert Weiner
I didn't have time to write an article, so I wrote a book instead.
-Kenneth Arrow

In the last two decades, digital microcontrollers and signal processors have become increasingly popular. Due to plummeting costs, their widespread applications now include cellular phones, automobiles, thermostats, robotics, and many, many others. Before the advent of digital systems, analog filters and controllers were the only option. However, the flexibility, reliability, repeatability, and low cost of current digital systems have made their analog counterparts relatively less common. Linear systems theory and its applications have also had to adapt to this newcomer, which introduced new problems and side-effects.

Since the early twentieth century, a vast amount of research and development has occurred in the theory of linear systems. Linear time-invariant (LTI) models very effectively describe much of the world that engineers are concerned with — small deviations from constant operating conditions — and have also been popular due to their relative analytical simplicity. Most controllers and signal processors are designed using LTI methods and, in a controller's case, are designed for an LTI plant model. However, digital implementations of linear systems are inherently non-linear due to the finiteness and consequent discretization of computer memory. Only a finite amount of precision can be dedicated to each number, and thus computations must be truncated after a finite number of digits. Furthermore, numbers can increase only until a finite limit, a limit that in practice may severely restrict the engineer. A system that is stable in infinite precision can become unstable when implemented in finite precision, for example displaying zero-input limit cycles where the system oscillates between non-zero states even when there is no input; more generally, overall performance differs from that of the infinite precision design.

Significant research on these finite precision effects in digital filters and, to a lesser extent, in digital controllers has given us many different methods to cope with these nonlinearities, and has allowed us to continue to use the powerful and well-developed LTI theory for the design of filters and controllers.

This thesis addresses the problem of minimizing FWL effects in fixed-point implementations of controllers and signal processors.

Previous Work

Traditionally, research on finite wordlength (FWL) effects, as they are commonly referred to, originated more in the digital signal processing (DSP) community, though Knowles and Edwards [93] developed some of the earliest seminal work on FWL effects in a controls (sampled-data) context. Around 1980, Moroney *et. al.* [119,120] also presented some ground breaking work in FWL effects in control systems. The last 15 years or so have seen an ever increasing interest in FWL effects in digital controllers and most recently, those involving hybrid systems (i.e. systems that combine both a continuous time (CT) and a discrete time (DT) or digital system into one).

Different aspects of the FWL problem in digital filtering have gained and receded in importance, depending on advances in technology. Minimizing roundoff errors and filter sensitivity have been constant concerns. [62,73,78,79,82,88,122,169] and many more works all address these topics. Two specific changes that have occurred recently:

(i) On most DSP chips, multiply instructions take the same number of cycles as many other instructions and are thus not the major contributors in the cost equation [68].

(ii) Advanced Very Large Scale Integration (VLSI) technology has made custom chip design far more popular. From that standpoint, different design criteria become more important. As early as 1984, Rao and Kailath [147] pointed out:

> "With modern technological advances, however, the criteria for designing digital filter realizations have undergone considerable transformation. In addition to classical criteria of low sensitivity with respect to finite word-length, absence of limit-cycle and overflow oscillations, etc., the realizations should at least also have the following desirable properties for VLSI implementation:
>
> a) The circuit should be of the concurrent array processor type i.e. a cascaded interconnection of identical processors with only nearest neighbor links. ...
>
> b) It should be pipelineable in order to maximize throughput, i.e. the circuit should be able to process input data at a rate that is independent of the order of the filter."

FWL effects vary significantly depending on the underlying numeric format. Most current microprocessors represent numbers in one of two formats, fixed-point or floating-point. While using the fixed-point format offers many advantages, it also exacerbates FWL effects. Thus, a lot of the research addresses FWL effects in fixed-point implementations.

What this thesis contains

My hope is to synthesize a framework to gather, organize, and interconnect past work while also accomplishing the following goals:

(i) to thoroughly analyze the digital filter and controller design process;

(ii) to expose obvious open questions and their importance;

(iii) to serve as a reference and as a catalyst for new work.

With this framework, a picture of current research should become relatively clear. Distinctions among different approaches should sharpen and similarities should become evident.

Using this thesis, an engineer should be able to make significant headway in answering the question, "How can I optimally implement this digital filter (or controller) using fixed-point hardware?" Implicit in this question is the need to first answer, "Optimal in what sense?" These two questions are the primary subjects of current and past research in this area, and this thesis comprehensively surveys this research.

While the explicit content should serve as a reference as well as an introduction to unfamiliar ideas, it is nonetheless the work of other scientists and engineers. My contribution is the actual organization of the material and the resulting coherent picture that should emerge as you read the thesis.

I will discuss many of the major works that have driven the field (such as [73, 120, 122, 169] and several others) and in some sense form a scaffolding for the overall framework of FWL research. To this, I will tie in much of the additional work. With each work, I will present a mathematical development of the main idea and also the underlying thoughts that may help to place it in a larger context.

I assume that you, the reader, have a basic background in dynamic systems, controls, and/or signal processing. Definitions of the terms used so far as well as those that follow and the notation are listed in the appendices. The second chapter has a basic presentation of finite wordlength representations. Throughout, references will guide you to more in-depth material as well. For the actual implementation phase, I point to some references (see for eg. [68]) to help select

- a digital signal processor,

- other hardware and software,

- and most importantly, the fixed-point or floating-point format.

The interdependence of design and implementation makes an independent discussion of either one impossible. For custom integrated circuits, design and implementation are intimately connected and can take advantage of the additional design freedom to assign different wordlengths to individual coefficients as well as to each data path between arithmetic components. Ideally, a software tool should rapidly explore the FWL digital filter and controller design spaces and, based on constraints, return several possible configurations.

What this thesis does not contain

While providing a lot of ideas in one place, I do *not* provide particular solutions. Ultimately, I leave the engineer to choose an approach and use it for design and analysis.

Furthermore, I do not include any significant discussion on limit cycles or scaling, the process of changing the range of the coefficients so that the system's internal variables stay within the dynamic range of the digital hardware. Section 2.3 has a short presentation on scaling as well as some references. For references on limit cycles, see [28,81,151]. I also develop everything in the context of fixed-point processing so floating point researchers may or may not find anything of use. Some authors have noted that optimal fixed-point algorithms generally perform well in the floating-point domain also [68, 146]. I do not discuss the choice between using fixed-point and floating-point either. Finally, I restrict this survey to digital controllers and one-dimensional digital filters. Gevers and Li discuss FWL effects in estimation and also include some additional references [62, chapter 11].

Thesis Organization

Chapter 2 discusses the fixed-point representation in some detail and presents the interaction between LTI theory and FWL effects. Chapter 3 contains one of the original contributions of this thesis: it describes, in extensive detail, the current digital filter and controller design process. It provides context for the remainder of the thesis and is a very important part of the framework that is ultimately my goal.

It also helps to organize the extensive list of references in the bibliography.

I sharply split the rest of the development along the lines implied in the question "What is the optimal realization for this system?" Chapter 4 describes many of the performance metrics[1] that define the interpretation of "optimal" while Chapter 5 presents the structures or realizations that optimize each metric. This organization separates the measure from the structure that minimizes it and also serves to highlight the distinction between *measuring* a structure's performance and its actual implementation. Some structures, while they may perform very well under the cost functions they were designed with, might be poor performers under other measures.

I have also opted, organizationally, to highlight two other distinctions, that between DSP and controls, and that between stochastic and deterministic measures. The *requirement* for the first distinction will become obvious in Chapter 3, while I chose the second distinction somewhat arbitarily to help organize the presentation.

Other Surveys

Each of the following sources gathers and presents many connected results:

- Liu's early survey in 1971 on FWL effects in digital filters [103];
- Moroney *et. al.'s* 1980 paper [120] and Moroney's 1983 book [119] on digital compensators;
- Hanselmann's 1987 survey of digital controllers [68];
- Butterweck *et al.*'s 1988 survey of FWL effects in digital filters (with an extensive bibliography of almost 450 references) [28, 29];
- Wilkinson's 1991 text on FWL considerations in controls [185];
- and Gevers' and Li's 1993 text [62].

[1] Throughout, I will use the term metric and measure interchangeably.

However, none are as broad in scope as this thesis. Moreover, exciting new developments, especially in the controls community in the last decade, have not been collected anywhere.

So, without further delay, onwards.

Chapter 2

Background

The first section of this chapter briefly describes the fixed-point numeric representation and should serve as a self-contained introduction and tutorial for reading the rest of this thesis. Section 2.2 discusses the intrusion of FWL effects on LTI theory. The third section contains a short discussion on scaling digital filters to reduce the probability of overflow.

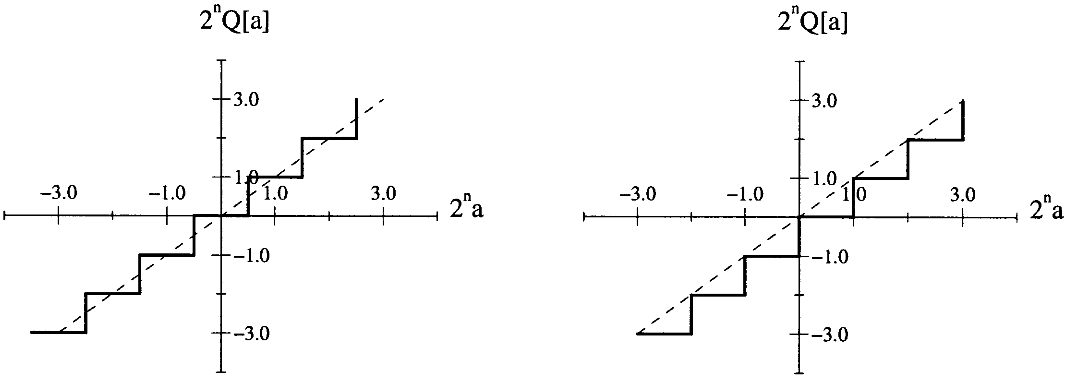
2.1 The Fixed-Point Representation

Computers use several different techniques for representing numbers internally. The most popular and common ones are fixed- and floating-point. Here, I describe the fixed-point representation in some detail while referring the interested reader to references for information on the others [68, 72, 167, 182, 185].

The two types of FWL effects are *quantization* and *overflow.*

Quantization

Quantization is the process of reducing a number a represented with more than n fractional bits down to one with only n fractional bits. A fractional bit appears after the binary point. Two common quantization methods are roundoff and truncation. Roundoff is the normal operation of rounding, while truncation simply discards all bits to the right of the *nth* fractional bit. Thus, $0.09375 (= 0_{\wedge}00011)$ rounded to four fractional bits would be 0.125($= 0_{\wedge}0010$) while truncated to four fractional bits it would be 0.0625($= 0_{\wedge}0001$). The resulting error due to quantization, $e = a - Q[a]$, is called the roundoff or truncation error. Characteristic curves for roundoff and truncation (in fixed-point two's complement) are shown in Figures 2-1(a) and 2-1(b).



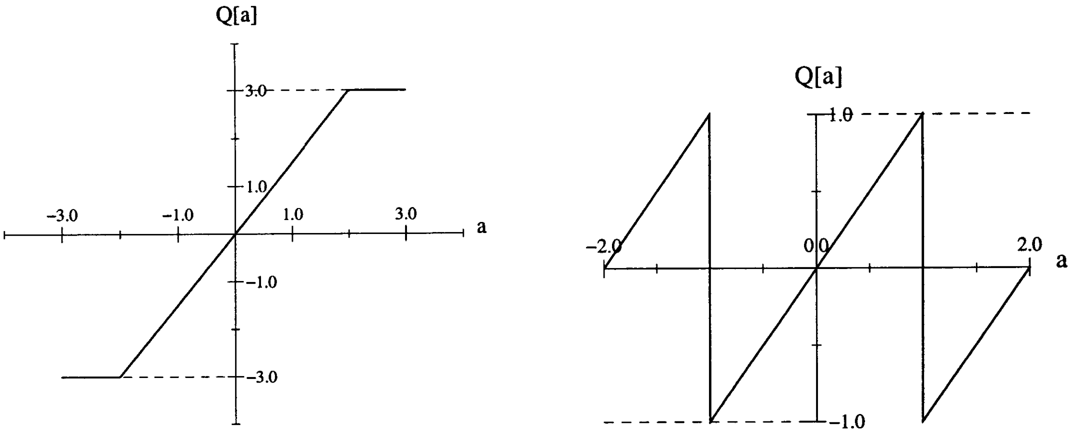
(a) Quantization plot using roundoff

(b) Quantization plot using truncation

Figure 2-1: Fixed-point quantization characteristics.

Overflow

An overflow occurs when the sum or product of two numbers is outside the dynamic range, the range of representable numbers. The two main methods of overflow handling are saturation and wrapping. The saturation characteristic's curve is shown in Figure 2-2(a). If the input is larger (smaller) than the largest (smallest) representable number, saturation returns the largest (smallest) representable number. The other technique, called wrapping, simply ignores overflows. For example, adding 1 (001) to 3 (011) in a 3-bit signed representation (with no fractional bits) would normally result in 4 which cannot be represented. Saturation would return 3 (011) while wrapping would return −4 (100). The characteristic curve is shown in Figure 2-2(b). While both these errors occur in all numeric representations, they take a slightly different form in each one.



(a) Saturation characteristic

(b) Wrapping characteristic

Figure 2-2: Overflow characteristics.

The Fixed-Point Representation

In fixed-point, a number a is represented as an integer part, a fractional part, and a sign bit. The wordlength w is the total number of bits used to store a. Say m bits are used for the integer part and n bits for the fractional part. Then, $w = m+n+1$, and a is stored as $a = a_m a_{m-1} ... a_{0\wedge} a_{-1} a_{-2} ... a_{-n}$ where a_m is the sign bit and $_\wedge$ represents the binary point. During any arithmetic operation, the location of the binary point remains constant and hence the term fixed-point. For example, adding two numbers does not require an operation to align the decimal places, which is usually necessary when adding two floating-point numbers.

Some of the various fixed-point formats are

(i) *Sign magnitude*

The first bit stores the sign while the remainder stores the magnitude:

$$a = (-1)^{a_m} \left(\sum_{i=0}^{m-1} 2^i a_i + \sum_{i=1}^{n} 2^{-i} a_{-i} \right) \tag{2.1}$$

(ii) *One's complement*

Positive numbers are stored as in the sign magnitude format while negative

	Sign Magnitude	One's Complement	Two's Complement
Range (min)	$-\sum_{-n}^{m-1} 2^i$	$-\sum_{-n}^{m-1} 2^i$	$-\sum_{nn}^{m-1} 2^i - 2^{-n}$
Range (max)	$\sum_{-n}^{m-1} 2^i$	$\sum_{-n}^{m-1} 2^i$	$\sum_{-n}^{m-1} 2^i$
+0	$0000_{\wedge}00$	$0000_{\wedge}00$	$0000_{\wedge}00$
−0	$1000_{\wedge}00$	$1111_{\wedge}11$	$0000_{\wedge}00$
3.75	$0011_{\wedge}11$	$0011_{\wedge}11$	$0011_{\wedge}11$
−3.75	$1011_{\wedge}11$	$1100_{\wedge}00$	$1100_{\wedge}01$

Table 2.1: Common numeric representations.

numbers are stored as *complements*. The complement operation, represented as an overbar, is defined as: $\bar{1} = 0$ and $\bar{0} = 1$.

$$a = \begin{cases} \sum_{i=0}^{m-1} 2^i a_i + \sum_{i=1}^{n} 2^{-i} a_{-i} & a \geq 0 \quad or \quad a_m = 0 \\ -\left(\sum_{i=0}^{m-1} 2^i \bar{a}_i + \sum_{i=1}^{n} 2^{-i} \bar{a}_{-i} \right) & a < 0 \quad or \quad a_m = 1 \end{cases} \tag{2.2}$$

(iii) *Two's complement*

Positive numbers are stored as in the sign-magnitude format, while negative numbers are stored in *two's complement*.

$$a = \begin{cases} \sum_{i=0}^{m-1} 2^i a_i + \sum_{i=1}^{n} 2^{-i} a_{-i} & a \geq 0 \quad or \quad a_m = 0 \\ -\left(\sum_{i=0}^{m-1} 2^i \bar{a}_i + \sum_{i=1}^{n} 2^{-i} \bar{a}_{-i} \right) - 2^{-n} & a < 0 \quad or \quad a_m = 1 \end{cases} \tag{2.3}$$

Two's complementing a number turns a into $-a$ and is carried out as follows: All 0's are switched to 1's and vice versa. 1 is added to the least significant bit (LSB) of the fractional portion of this result.

The dynamic range for all three formats spans approximately from -2^m to 2^m. Table 2.1 summarizes these properties and lists some examples. Each representation has its advantages and disadvantages, but since most digital signal processors and controllers (as well as most general purpose computers) use two's complement, this thesis will focus on it.

Two's complement has a unique representation of zero and is also immune to overflow errors during a series of additions and subtractions, as long as the final result is within the dynamic range. An example quickly illustrates the point. Consider an example in a 4-bit signed two's complement format (with no fractional part). The left column indicates the correct sum, while the right column indicates the interpretation of the sum in two's complement.

$$\begin{array}{rrr}
+5 & +5 & 0101 \\
+4 & +4 & 0100 \\
\hline
+9 & -7 & 1001 \\
+7 & +7 & 0111 \\
\hline
+16 & 0 & 0000 \\
-2 & -2 & 1110 \\
\hline
14 & -2 & 1110 \\
-8 & -8 & 1000 \\
\hline
+6 & +6 & 0110
\end{array}$$

Subtraction is easily implemented as two's complementation followed by addition.

Advantages and Disadvantages of the Fixed-Point Representation

The popularity and importance of the fixed-point format attest to its many advantages. Fixed-point chips are cheaper, run faster and cooler, and take less space to implement a certain level of performance than their floating point counterparts. They dominate the markets for mass-market applications like cellular phones and where speed is of extreme importance (e.g. in high-performance real-time systems).

Coincidentally, the additive roundoff error in fixed-point quantization yields much more easily to analysis than the multiplicative roundoff error of the floating-point format.

The primary disadvantage of using the fixed-point format is its higher sensitivity to FWL effects and its limited dynamic range.

A finite dynamic range also requires a technique to handle overflows. Fixed-point representations, when compared to floating point, have an especially limited dynamic range and thus are much more likely to have overflows. Concerns about overflow handling and reducing its probability of occurrence enter significantly into roundoff error models and limit cycle analysis.

To extend the dynamic range of the fixed-point format while maintaining its advantages, Wilkinson proposed a dynamically scaled fixed-point format; see [185, chapter 2] for details.

2.2 FWL Effects and LTI Systems

This section describes how FWL effects enter the picture and affect LTI models. Let's start with the standard discrete-time state-space description,

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] + Du[k] \end{aligned} \tag{2.4}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}, D \in \mathbb{R}^{p\times m}$, $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector, and $y \in \mathbb{R}^p$ is the output vector. The notation (A, B, C, D) or $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ compactly describes the system.

Notice the many places where FWL effects enter. First, each entry of the coefficient matrices must be rounded. Then, the input $u[k]$ must be rounded also. Finally, the result of each state update, $x_i[k+1] = \sum_{j=1}^{n} a_{ij}x_j[k] + \sum_{j=1}^{m} b_{ij}u_j[k]$, needs to be rounded.

Let's introduce the effects one at a time.

Coefficient Quantization

First, we round the coefficients in the system matrices. Let (A^*, B^*, C^*, D^*) be the system matrices with each entry rounded to some precision:

$$\begin{aligned} x^*[k+1] &= A^*x^*[k] + B^*u[k] \\ y^*[k] &= C^*x^*[k] + D^*u[k] \end{aligned} \tag{2.5}$$

where

$$\left[\begin{array}{c|c} A^* & B^* \\ \hline C^* & D^* \end{array}\right] = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] - \left[\begin{array}{c|c} \Delta A & \Delta B \\ \hline \Delta C & \Delta D \end{array}\right] \tag{2.6}$$

In the most general case (where one can control each data path for each multiply and add), one could round each entry to a different precision depending on its importance (and that of the corresponding state) in the final system. In Sections 5.6 and 5.10, I will describe work that exploits this optimization. To initially simplify the presentation, I will assume that all entries are rounded to B_c fractional bits. Each entry of $(\Delta A, \Delta B, \Delta C, \Delta D)$ will thus lie in the range $[\frac{-2^{-B_c}}{2}, \frac{2^{-B_c}}{2}]$.

Note that $x^*[k]$ does *not* mean the quantized state but rather the *infinite precision* state evolving with the quantized system matrices. Thus, the system remains linear. The rounded coefficients change the system matrices and properties but do not affect linearity. Stability can be deduced by computing the eigenvalues of the new A^* matrix.

The error equations evolve as

$$x_e[k] = x[k] - x^*[k] \qquad\qquad y_e[k] = y[k] - y^*[k]$$

Substituting from (2.5) and (2.6),

$$\begin{aligned} x_e[k+1] &= Ax[k] + Bu[k] - (A^*x^*[k] + B^*u[k]) \\ y_e[k] &= Cx[k] + Du[k] - (C^*x^*[k] + D^*u[k]) \\ x_e[k+1] &= \Delta Ax[k] + A^*x_e[k] + \Delta Bu[k] \\ y_e[k] &= \Delta Cx[k] + C^*x_e[k] + \Delta Du[k] \end{aligned} \tag{2.7}$$

Writing (2.7) as a MISO[1] system driven $x[k]$ and $u[k]$, one can measure the frequency and step response of the output error.

The transfer function description

$$H(z) = \frac{N(z)}{D(z)} = \frac{b_0 + b_1 z + \ldots + b_m z^m}{a_0 + a_1 z + \ldots + a_n z^n}$$

[1] Following convention, SISO denotes a single-input/single-output system, while MIMO is multi-input/multi-output. SIMO is single-input/multi-output, while MISO is muti-input/single-output.

will instead be realized as

$$H^*(z) = \frac{N^*(z)}{D^*(z)} = \frac{b_0^* + b_1^* z + ... + b_m^* z^m}{a_0^* + a_1^* z + ... + a_n^* z^n}$$

where again,

$$\Delta H(z) = H(z) - H^*(z)$$

$\Delta H(z)$'s frequency and step response are also easily measured as it is simply the sum of two linear systems. Assuming small coefficient perturbations, a first order approximation of $\Delta H(z)$, useful in succeeding chapters, is

$$\Delta H(z) = \sum_{i=1}^{n} \frac{\partial H(z)}{\partial c_i} (\Delta c_i) \,|_{c_i}$$

where the c_i are the coefficients in $H(z)$ and Δc_i is the perturbation in c_i.

Hinting at another possible approach to analyze this problem, one could model coefficient quantization as an additive perturbation of the controller or filter (see Figure 2-3). I will return to this approach in Section 4.10.

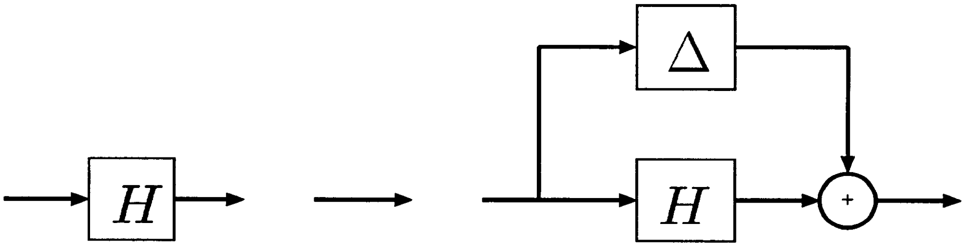


Figure 2-3: Coefficient quantization modeled as an additive perturbation.

State and Input Quantization

Incorporating state and input quantization changes the model (2.5) to

$$\begin{aligned} x_Q^*[k+1] &= A^* Q[x_Q^*[k]] + B^* Q[u[k]] \\ y_Q^*[k] &= C^* Q[x_Q^*[k]] + D^* Q[u[k]] \end{aligned} \qquad (2.8)$$

where $Q[\cdot]$, the quantization operator, reduces its operand's fractional wordlength to B_s bits. Here, $x_Q^*[k]$ represents the quantized state. The error equations now become

$$x_e[k+1] = x[k+1] - x_Q^*[k+1]; \qquad y_e[k] = y[k] - y_Q^*[k]$$

$$x_e[k+1] = Ax[k] + Bu[k] - (A^*Q[x_Q^*[k]] + B^*Q[u[k]])$$

$$y_e[k] = Cx[k] + Du[k] - (C^*Q[x_Q^*[k]] + D^*Q[u[k]])$$

which, due to the non-linearity of the quantization operator, are non-linear.

2.3 Scaling

As mentioned before, the fixed-point format severely restricts dynamic range. Thus, the size of internal numbers must be kept relatively small to avoid overflow and provide accurate computations. Typically, each internal variable is scaled (i.e. multiplied by a suitable number) so that it stays within the dynamic range.

In general, if f is the impulse response sequence from the input to an internal variable, v, then $v[k] = (f{*}u)[k]$ or equivalently, $V(e^{j\omega}) = F(e^{j\omega})U(e^{j\omega})$ where $u[k]$ is the input sequence. The variable v can be any of the states in the system. Since the variable's value depends on the input, scaling also depends on the class of inputs the system will operate on and the importance of preventing overflow. Each different norm applied to the input and the system's resulting impulse response sequence determines a different scaling rule. Table 2.2 from Roberts and Mullis' textbook [151, page 365] lists the range of the internal variables depending on the input and the appropriate norm.

Using these inequalities, one can write some scaling rules:

$$l_1 \text{ scaling}: \qquad ||f||_1 = \sum_{k=0}^{\infty} |f[k]| \qquad = 1 \tag{2.9}$$

$$l_2 \text{ scaling}: \qquad ||F(z)||_2 = \left[\sum_{k=0}^{\infty} (f[k])^2\right]^{1/2} = 1 \tag{2.10}$$

$$l_\infty \text{ scaling}: \quad ||F(z)||_\infty = \max_{\omega} |F(e^{j\omega})| \quad = 1 \tag{2.11}$$

Input Class	Range of internal variables ($v[k]$)
Bounded inputs $\|u[k]\| \leq 1$	$\|v[k]\| \leq \sum_l \|f[l]\| = \|\|f\|\|_1$
Finite energy inputs $\sum_{l \leq k} (u[l])^2 \leq 1$	$\|v[k]\| \leq [\sum_l (f[l])^2]^{1/2} = \|\|F(z)\|\|_2 = \|\|f\|\|_2$
Sinusoidal inputs $u[k] = cos[k\omega]$	$\|v[k]\| \leq \max_\omega \|F(e^{j\omega})\| = \|\|F(e^{j\omega})\|\|_\infty$
White inputs $S_{uu}(e^{j\omega}) = 1$	$[E[(v[k])^2]]^{1/2} = \|\|f\|\|_2$
Wide sense stationary inputs $r_{uu}[k] \leftrightarrow S_{uu}(e^{j\omega})$	$[E[(v[k])^2]]^{1/2} = \left[\frac{1}{2\pi}\int_0^{2\pi} S_{uu}(e^{j\omega})\|F(e^{j\omega})\|^2 \, d\omega\right]$

Table 2.2: Range of internal variables $v[k] = (f * u)[k]$ for various classes of inputs u.

These rules assume zero initial conditions in the filter and neglect roundoff noise error inputs.

The relations $||u||_\infty \leq ||U||_1 \leq ||U||_2 = ||u||_2 \leq ||U||_\infty \leq ||u||_1$ imply that the l_1 bound is the most conservative. Often, the l_2 scaling rule is used in a modified form with a factor α:

$$\alpha ||F(z)||_2 = \alpha \left[\sum_{k=0}^{\infty} (f[k])^2 \right]^{1/2} = 1 \tag{2.12}$$

where α can be interpreted as the number of standard deviations representable with the finite wordlength available, assuming a unit-variance white noise input.

The choice of a scaling rule represents a tradeoff between dynamic range (overflow errors) and roundoff errors. For example, the very pessimistic l_1 scaling rule *guarantees* no overflows, i.e. for any bounded input, the probability of overflow is 0. However, that means that most of the numbers will be *very* small and the roundoff error after multiplication will be a significant percentage of their magnitude. For example, with three fractional bits, 0.021 × 0.021 = 0.000441 would be rounded to 0 with a rounding error of .000441 which is about 2%; 0.21 × 0.21 = 0.0441 would be rounded to 0.044 with a rounding error of 10^{-4} which is about 0.05%! Of course, some choices of numbers may cause the reverse to be true, but in general, this example is representative. In most cases, an l_1 scaling wastes a lot of the dynamic range and increases roundoff error, making modified l_2 scaling the most common choice; [151] states that $\alpha > 5$ is considered conservative.

In the controls context, one would also have to include the plant model in determining scaling rules; see [119, 185] for extensive discussion of scaling rules, especially the l_2 rule, in the controls context.

The scaling rule is actually implemented so as to leave the overall transfer function unchanged. Figure 2-4 depicts the transfer function from u to y with an internal variable v and $\beta = \|f\|_1, \|f\|_2$, or $\|F(e^{j\omega})\|_\infty$ depending on the scaling rule chosen. Note that the scaling rule will also change for MIMO systems.

$$\begin{tikzcd} u \arrow[r, "F"'] & v \arrow[r, "G"'] & y \arrow[r, thick] & u \arrow[r, "\frac{F}{\beta}"'] & v \arrow[r, "\beta G"'] & y \end{tikzcd}$$

Figure 2-4: Scaling implementation.

I will discuss scaling in the state-space representation in more detail in Section 4.5.

One final note: In many practical cases, scaling is actually done by simulation. The filter or controller is simulated with the different input signals that it is expected to receive during operation, and the maximum of the states during operation determines the scaling. See [166] for a description of software that facilitates the use of this method.

2.4 Summary

So far, I have discussed the repercussions of the fixed-point representation on general LTI theory. The error models here will serve as a starting point for further developing the analysis in each succeeding section.

The next chapter presents a larger picture of the FWL design process i.e. the process of starting with a CT plant and designing a FWL digital controller for it or that of designing an FWL digital filter. It will provide context for the succeeding presentation and will also help to categorize the large amount of literature on this topic.

Chapter 3

Controls and Signal Processing: Differences and Similarities

... the "separation principle", which is at the heart of much of the linear state estimate feedback control theory, whether it be by pole placement or by LQG control, breaks down in practice at the implementation phase of the control law. This is a key observation that should haunt the nights of many linear or linear quadratic control theoreticians.

– Gevers and Li in [62]

Researchers and engineers, for the past three or more decades, have been designing digital controllers and filters that account for FWL effects and try to minimize them. The mathematical and analytical tools and the computational resources at their disposal have shaped their different views and approaches to the problem. This chapter describes the design processes, both for digital filters and for digital controllers. The first section gives a brief outline while the second gives a much more detailed description. This design process is a major part of the framework that this thesis lays out and provides a natural organization for many of the references also. Figure 3-4 is an annotated figure showing both the filter and controller design processes.

3.1 The Design Process: A Summary

Filter Design

In the DSP context, the design path usually involves only one step: from a DT filter to an FWL DT filter. Sometimes, it involves a prior step, discretizing a CT filter to a DT filter (see Figure 3-1).

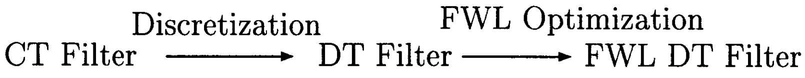


Figure 3-1: FWL DT filter design.

Controller Design

Including the CT plant and a feedback loop fundamentally distinguish the controller problem from the filter problem. The goal in controls is to optimize the performance of the *entire* closed loop, not just that of the controller. In the controls context, a fork marks the longer design path (solid lines in Figure 3-2).

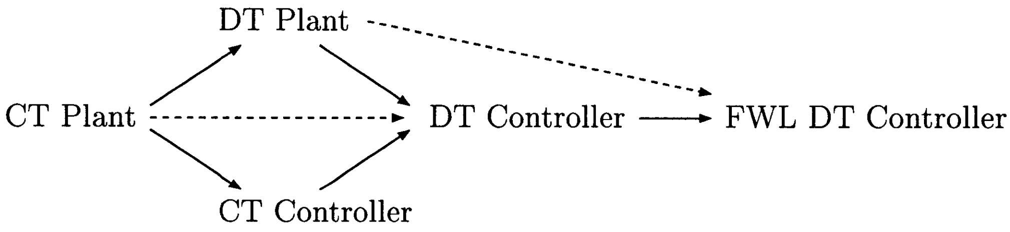


Figure 3-2: A summary of the FWL design process.

Why not a direct path from the CT plant to an FWL DT controller? Or at least to a DT controller? The answer to the latter question is that most traditional controller design methods deal with *either* continuous- *or* discrete-time, but not both. Starting with a CT plant, one can either discretize the plant or design a CT controller (Figure 3-2).

If the CT domain is chosen, one proceeds as follows:

1) First, a CT controller is designed using one of many standard techniques. One advantage of doing CT controller design is the ease of specifying and translating

performance requirements.

2) Once designed, the CT controller is discretized using one of several methods, with sophistication varying from the simple bilinear transformation to Keller and Anderson's discretization method [92] (Section 5.9.3) that takes the CT plant into account.

Algorithms that perform step (2) above may be further categorized into

(a) those that do not take the plant into account, and

(b) those that take the plant, and hence the entire closed-loop behavior, into account.

For example, a digital redesign technique that matches the states of the DT controller to those of the CT controller at the sampling instants would fall into the first subgroup. A better redesign method that matches the states of the CT and DT *closed-loop* systems at the sampling instants (see e.g. [58]) would fall into the latter subgroup.

If the DT domain is chosen, one proceeds as follows:

1) First, the plant is discretized at the controller's sampling frequency.

2) Then, linear quadratic Gaussian (LQG), H_∞ design, or any other method is used to compute a DT controller.

This route has two disadvantages:

- The entire process must be repeated if the sampling frequency changes.
- Some of these design methods only take plant and output behavior into account at sampling instants. Thus, they would treat the systems generating the step responses in Figure 3-3 as equivalent. Clearly, one is preferable.

Following either path, the DT controller must finally be optimized for FWL implementation. Again, FWL discretization can ignore the plant and treat the controller as a filter, or it may take closed-loop performance into account.

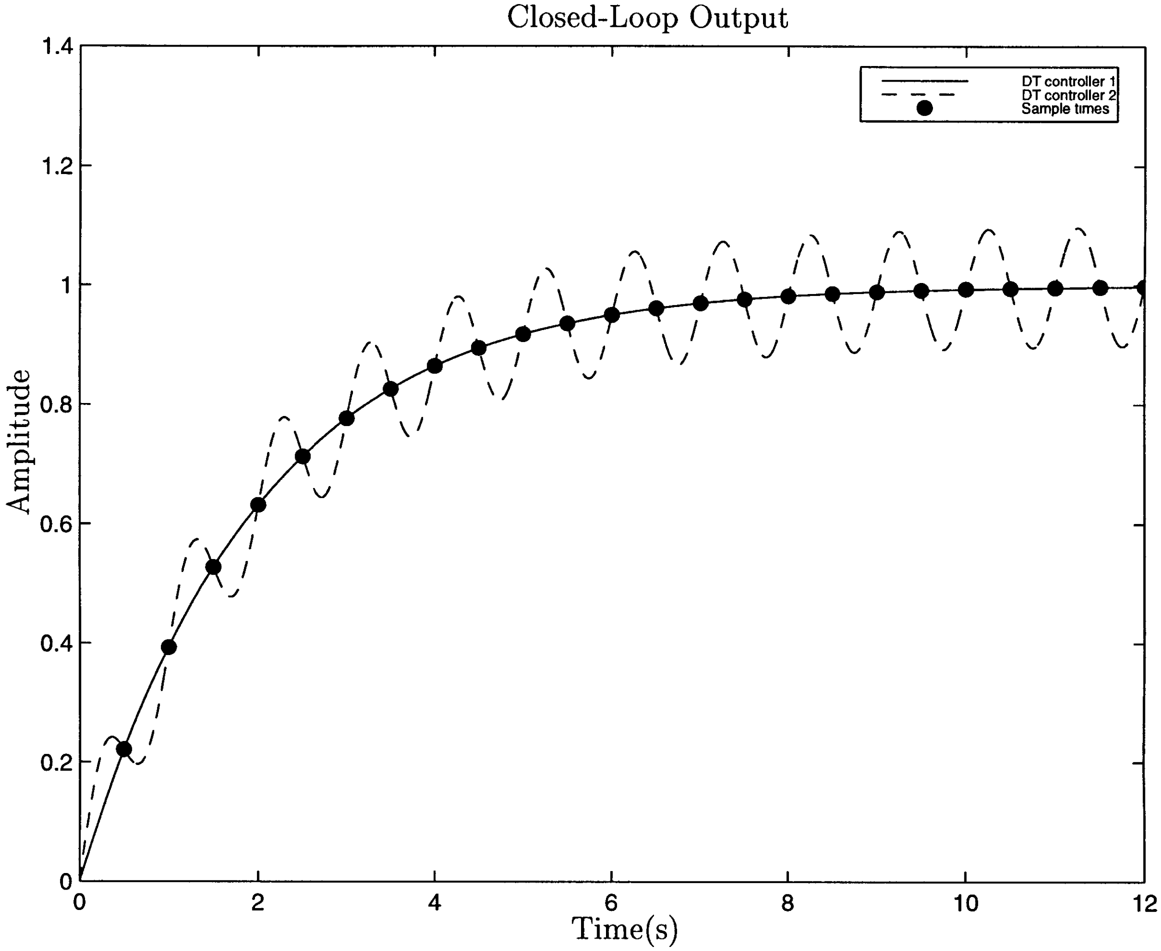


Figure 3-3: An example of intersample ripple.

The preceding discussion summarizes the "standard" design methods. Recent publications (in the last decade) have provided additional options and shortcuts (indicated by dashed lines in Figure 3-2) in the design methodology. Sampled-data H_2/H_∞ controller design methods now allow direct design of a DT controller in closed-loop with a CT plant [8,36,37,46,47,165], leaving only the FWL optimization step. Another bypass, due to Liu *et al.*'s work [105], allows direct FWL DT controller design based on a DT plant (using a minimum LQG cost criterion).

Similarities in the design processes

In both filter and controller design, the different discretization methods fall into one of two categories:

(i) those that search for the set of optimal similarity transformations for a given system, to minimize a given measure.

(ii) those that actually search over all possible realizations to find the set of systems (controllers or filters, depending on the respective problem) that best meets all the design constraints.

Mathematically, the first problem is

$$\min_{\{T \mid \det T \neq 0\}} M(T^{-1}AT, T^{-1}B, CT, D)$$

where (A, B, C, D) are fixed (determined by some other method) and T is a similarity transformation. The second problem's statement is

$$\min_{(A,B,C,D)} M(A, B, C, D)$$

$M(\,\cdot\,)$ denotes a performance measure in both cases. The latter, more difficult problem requires a search over a much larger set of possible solutions.

Design of feedback and observer gain matrices perfectly highlights the difference. Earlier solutions generally followed the three-step path [62] below:

1) plant discretization;

2) controller design via solution of (infinite-precision) DT Riccati design equations;

3) FWL discretization by finding the optimal transformation to minimize a performance measure (such as LQG cost).

Liu *et al.* [105] describe a two-step approach:

1) plant discretization;

2) direct computation of the optimal FWL DT controller by solving DT LQG design equations that take roundoff noise into account.

Thus, they solve *different* design equations than those in the infinite-precision case (see Section 5.10 for more details on Liu *et al.*'s method).

The rest of this chapter explains the design process in more detail and compares it to some of those described by other authors.

3.2 The Design Process: A Bigger Picture

Figure 3-4 shows the current FWL digital filter and controller design procedure in finer detail. I will first describe this figure and the associated design process (already summarized above) and then discuss how others authors have outlined the design process.

Filters and Filtering

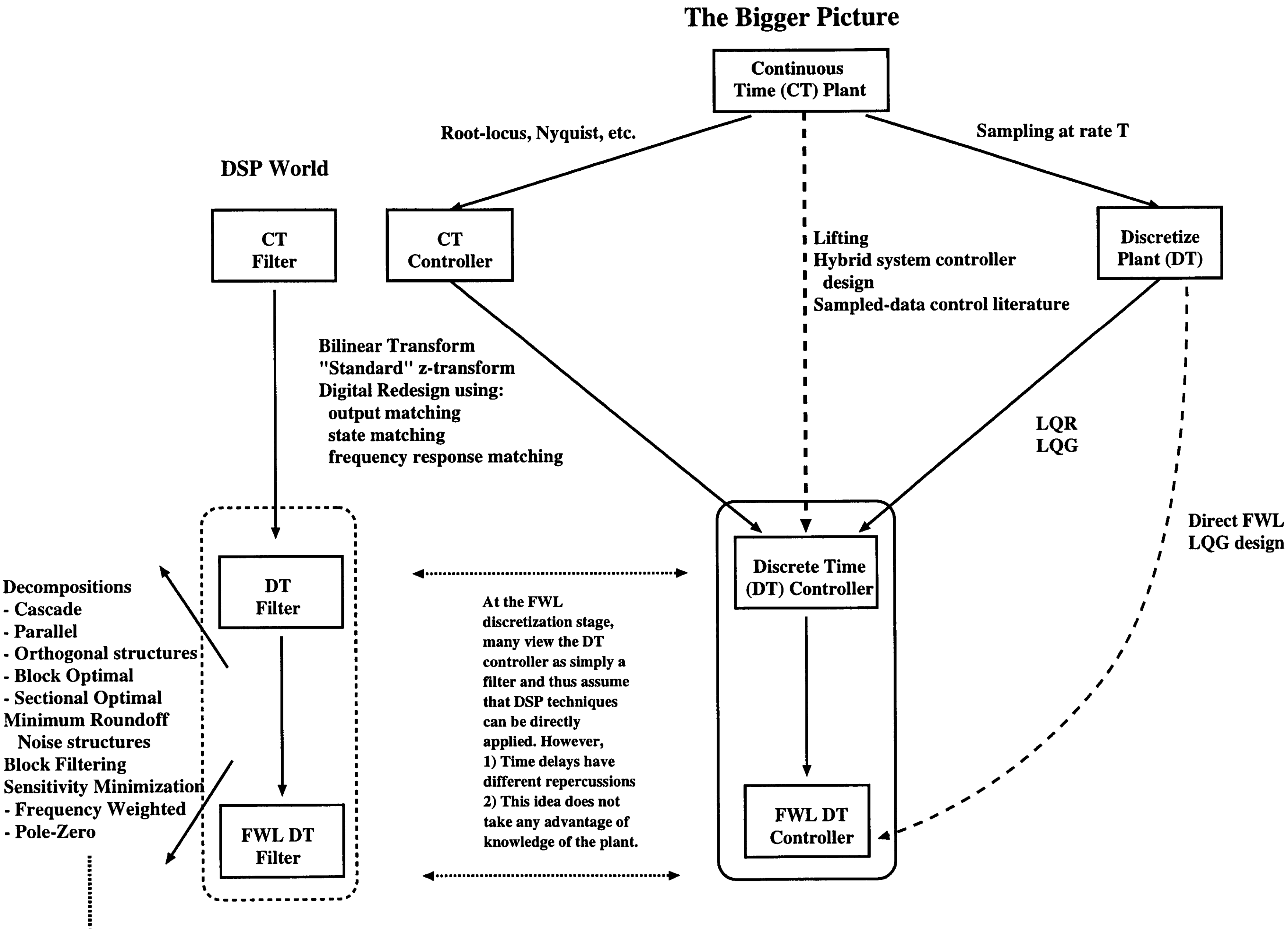
The relatively straightforward digital filter design process has been explored in much more depth than the corresponding digital controller design problem. If the process starts with a CT filter, then it proceeds down the path of discretization followed by FWL error minimization (Figure 3-5). For the many existing DT filters, one can simply apply the last stage of optimization for FWL performance. One could also use H_2/H_∞ design methods to minimize the error between a CT filter and a DT filter. The CT filter can be approximated using fast sampling followed by lifting (an approach developed for CT controllers in [92]). This approach may have already been explored in the DSP literature using the idea of blocking.

One avenue that past research has not explored is shown in Figure 3-6 — a direct route from a CT filter to an FWL DT filter. The design goal would be to minimize some measure of the output error.

Control, Controls, and Controllers

Control is generally done for CT or analog plants, so controllers are typically based on a CT (usually LTI[1]) model of the plant. The development of control theory, since the early '40s with the pioneering work of Bode, Black, Nyquist, and many others, to the present, has provided many tools to design CT controllers. Current design criteria allow controller and closed-loop specifications in the time and frequency domain. These specifications usually follow intuition and have a physical interpretation, and

[1] The remaining discussion will assume an LTI model



At the FWL discretization stage, many view the DT controller as simply a filter and thus assume that DSP techniques can be directly applied. However,
1) Time delays have different repercussions
2) This idea does not take any advantage of knowledge of the plant.

Figure 3-4: The digital filter/controller design process.

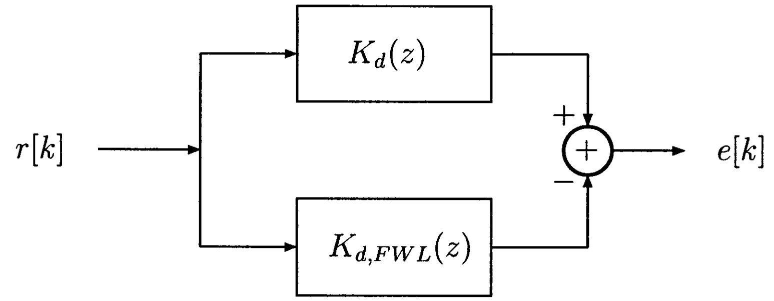


Figure 3-5: FWL DT filter error.

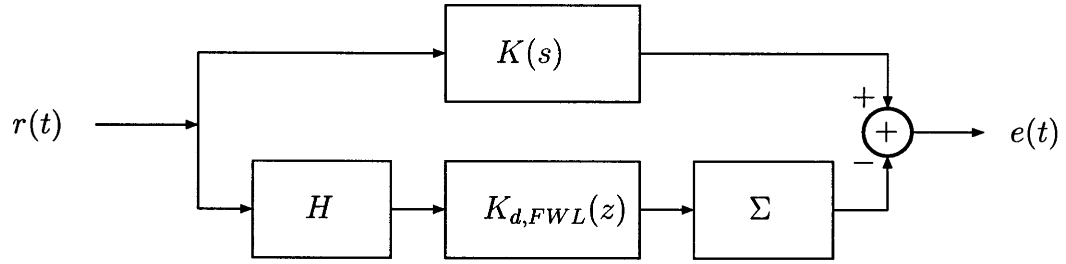


Figure 3-6: FWL DT filter error.

consequently are easy to understand. Thus, the CT plant → CT controller route is very popular.

The advent of cheaper, more reliable, repeatable and flexible digital technology has prompted a shift towards DT controllers and FWL DT implementations of them. What are the existing design methods that ease the shift from CT to DT controllers? There are many of these methods, generically called discretization methods. The categories are (as in the summary):

(a) those that take do not the plant into account;

(b) those that take the plant, and hence the entire closed-loop behavior, into account.

Methods in category (a) ignore the plant and the feedback loop, and treat the controller like a filter. Thus, DSP discretization techniques all directly apply. There are many such techniques, and most basic texts on signal processing cover several of them [24, 130]. Since much of the research on discretization orginated in the DSP community, many digital controllers were initially designed and implemented like filters. However, clearly stating the goal of discretizing a controller will indicate the suboptimality of this approach: Discretization should result in a controller such that the performance of the *entire* closed-loop matches that of the closed-loop system with the CT compensator (Figure 3-9). Thus, the design methods of category (b) are far more appropriate for controller design. Of course, performance (or minimum error) can be measured in many different ways.

If designing a DT controller from scratch, as opposed to redesigning a proven CT controller to function in DT, one has the option to pursue a different path, that of discretizing the plant first. Typically, the plant is discretized at the same sampling rate as the controller. Thus, the closed-loop system is entirely in discrete-time, and one can use standard DT methods that, given closed-loop DT performance criteria, return the optimal DT controller.

In either case, the DT controller must now be implemented on a DSP or a digital controller and thus in finite precision. Floating-point implementations, especially in modern high-end chips, generally approximate infinite-precision computation well, so one can implement the DT design immediately. However, when implemented in fixed-point, especially in small, low-cost, fixed-point chips, FWL effects are not negligible, and the DT design must usually be modified to mitigate these effects. This is the final step in the design process.

Again, the FWL discretization methods fall into two separate categories as before. However, now, we can break each category down further:

(a) those that take do not take the plant into account:

    (i) those that search for the set of optimal similarity transformations for a given measure and a given system;

    (ii) those that actually search over all possible realizations to find the set of systems that best meet all the design constraints.

(b) those that take the plant, and hence the entire closed-loop behavior, into account:

    (i) those that search for the set of optimal similarity transformations for a given measure and a given system

(ii) those that actually search over all possible realizations to find the set of systems that best meet all the design constraints.

The same warning that applied earlier for category (a) design methods is still in order when used with controllers: Digital controllers should *not* be treated as digital filters. If they are, the optimal FWL discretization minimizes the error (measured by some metric) in Figure 3-5. As shown in the categorization above, the algorithms to do this minimization can be separated into two groups.

The methods in category (b) offer more implementation options and are (as before) more appropriate to controls problems. Figures 3-7, 3-8, and 3-9 show some of the possibilities. Figure 3-7 minimizes the error between a DT closed-loop system with an infinite-precision DT controller and one with an FWL DT controller. A (b)-(ii) method controller design would perform better [105, 196]. An even better solution is to *not* discretize the plant and to minimize the output error as in Figure 3-8 The only current method that does this is in [111] and it follows (b)-(i). If we want FWL discretization to be closer to the earliest design stages, the ideal solution would be as in Figure 3-9 — an FWL system that performs as closely as possible to an ideal CT controller and plant combination.

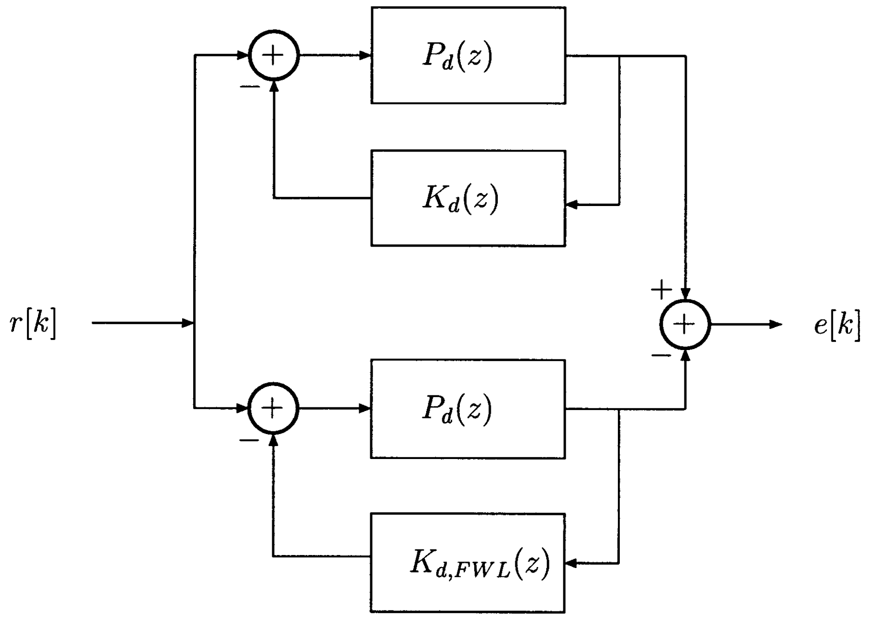


Figure 3-7: FWL DT controller error.

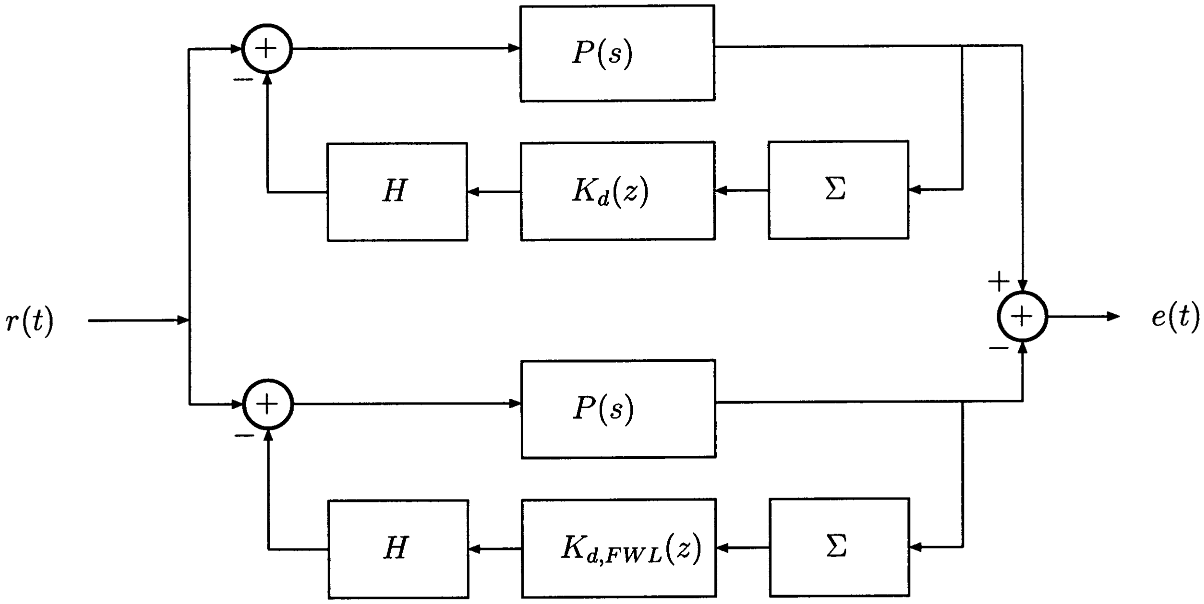


Figure 3-8: FWL DT controller error.

Other views of the design process

I will now discuss how three other sets of authors have described the design process. Hanselmann's survey of digital controllers [68] discusses discretization and filter structures. Discretization techniques all fall under the CT controller → DT controller branch. Hanselmann interestingly points out that the CT controller → DT controller route may be better than CT plant → DT controller. The latter requires specification of sampling frequency and computational delay and possibly other parameters, none of which may be known until an actual implementation is carried out. Parameter changes at that stage will necessitate a complete redesign. However, I would argue that choosing the CT controller → DT controller route will still require a rediscretization due to parameter changes, and adequate testing of the new design. Hanselmann also discusses different filter realizations for the controller, but never ties this in explicitly with the discretization step above. His survey however includes an excellent discussion about actually implementing digital controllers including some discussion of signal processing and controller hardware as well as software issues. He also discusses pipelining and time-delay issues in controllers.

Gevers and Li [62] describe the following design hierarchy progressing from the

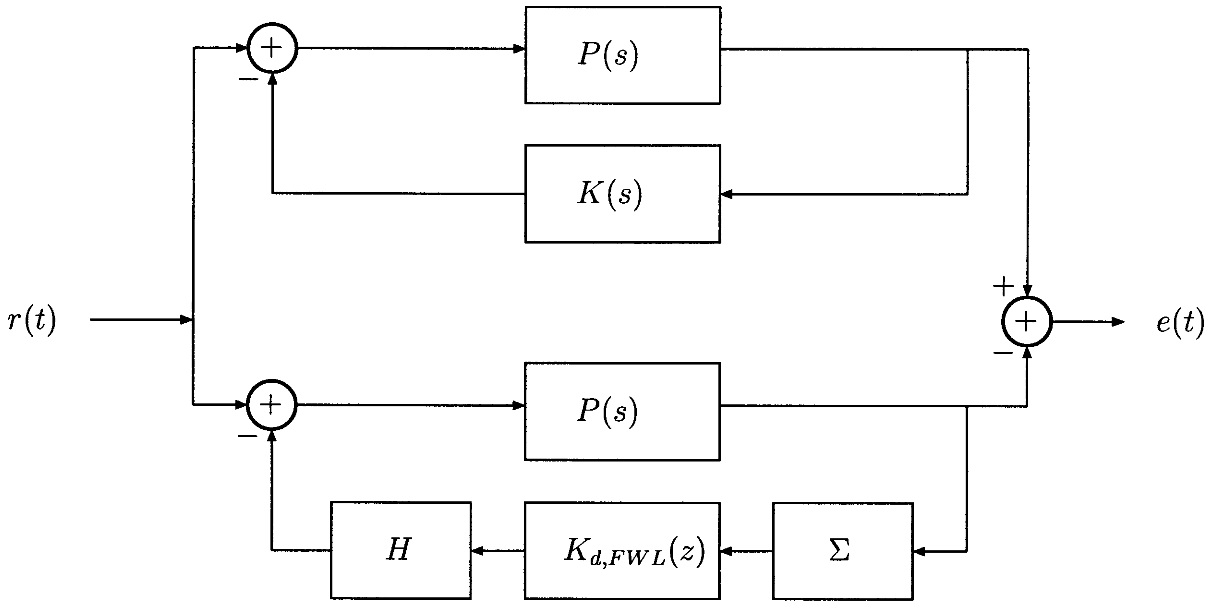


Figure 3-9: FWL DT controller error.

least favorable choice to the most favorable (the first three strategies assume a discretized plant):

1. Compute an ideal DT controller using any method, and then apply an FWL filter design criterion, treating the controller as a filter. This strategy corresponds to Figure 3-5.

2. Compute an ideal DT controller and then optimize the implementation by searching for the similarity transformations that best match its behavior to the ideal closed loop system's behavior, where 'best' can be measured with any of the performance metrics mentioned in the next chapter. This strategy corresponds to Figure 3-7.

3. Optimize the FWL controller over all FWL controller realizations with the given wordlength. Differs from (2) in that the controller transfer function itself is modified, not just its implementation. This is the same difference as that between the two cases mentioned in my description of the FWL discretization stage. This strategy also corresponds to Figure 3-7 but searches over a larger parameter space.

4. Optimize the FWL controller by comparing its performance with that of an infinite precision DT controller, both in closed loop with the continuous time plant itself, not with a discrete time approximation of it. This strategy also corresponds to Figure 3-8.

In each case, performance can be measured with one of many different metrics.

Chen and Francis [36] describe the following approaches:

- Analog design followed by discretization (the CT Controller → DT Controller route)
- Plant discretization followed by DT design (the CT Plant → DT Plant route)
- Direct DT controller design using sampled data methods (the direct CT Plant → DT Controller route)

Chen and Francis however do not describe the implementation stage at all, the DT Controller → FWL DT Controller.

3.3 Implementation Details

There are four implementation details that I did not see discussed much (if at all) in the FWL literature:

1) optimal wordlength allocation for individual registers;

2) generalized-hold functions;

3) sampling rate;

4) sampling resolution.

The first of these is most relevant (or only relevant) in custom integrated circuit design. The problem is to determine how many bits should be allocated to each coefficient, to each state variable, and to each adder and multiplier. This *very non-linear* and difficult question has received little attention. The only relevant works that I came across were:

- Roberts and Mullis [151] discuss optimal wordlength for minimum roundoff noise. They solve the following problem: Given a total number of bits B, how to optimally allocate them among the state variables to minimize the roundoff noise gain.

- Liu *et al.* [105] and Zhu *et al.* [196] both discuss optimal allocation of bits for state variables in LQG design, and their design algorithms output wordlength. Also, Williamson and Kadiman [186] importantly mention that reducing the number of bits allocated to a state is a way to do *partial* order reduction.

- Some authors have turned to general non-linear optimization techniques such as genetic algorithms and simulated annealing to facilitate the search for an optimum. They include the wordlength of each state in the cost function. Such optimization methods are completely flexible and can incorporate almost any criteria into their cost functions. However, as the search space gets more complicated, convergence will take longer and local maxima and minima are more likely to fool the search algorithm. [31] and [166] describe software that implements simulated annealing-based searches to minimize the wordlength for coefficients and states. Sung and Kum's algorithm [166] has been implemented in a commercial product, Cadence Design's Fixed-Point Optimizer^{tm}.

Generalized-hold hardware allows the designer more control freedom than the standard zero-order-hold. For an excellent discussion of generalized sample-and-hold hardware, refer to Araki's 1993 survey [6]. Additionally, [53] gives a detailed introduction to generalized-hold hardware as well as examples of its use in sampled-data systems.

I could not investigate the role of sampling rate and sampling resolution in FWL effects, and leave it for future work.

Chapter 4

Performance Metrics

To measure how bad a certain side effect is, we can equivalently measure how well the overall system performs in the presence of that side effect. Either way, how does one measure "optimality"? What cost function should we minimize? Many criteria vie for candidacy. For example, the typical ones of minimizing roundoff noise or coefficient perturbation sensitivity may be the clear choices. Or one could additionally include the hardware cost of implementing the particular realization. Latency restrictions or throughput requirements may also complicate matters. These latter criteria enter very non-linearly into cost functions and usually are not directly (or adequately) included in models. Thus, "optimal" FWL implementations still remain an art. In this chapter, I will discuss several performance metrics, their inter-relationships, and similarities and tradeoffs amongst the different ones.

Usually performance analysis for each FWL effect — coefficient quantization and state roundoff errors — is distinct, with a different measure for each. Several authors, though, have pointed out relationships between them, and Gevers and Li [62] actually give a synthetic measure that unifies the measurement of both effects into one number.

The coefficient quantization problem has been tackled with both stochastic and deterministic approaches, while a stochastic model has been applied for state quantization and multiplicative roundoff error.

Maintaining the organization mentioned in the Introduction, the sections in this chapter are organized with the DSP work first and then the sections on controls.

Within each major category, I first describe the deterministic measures and then the stochastic ones. I chose this latter distinction (somewhat arbitrarily) to help organize the presentation.

DSP - Deterministic Measures

A specific realization determines coefficient errors exactly, and one should be able to analyze their effects deterministically. The mathematical tools of the deterministic approach are sensitivity functions and their norms. Since quantization perturbs each coefficient by a small amount, the theory of polynomial coefficient perturbations and matrix perturbations has been extensively used to study quantization effects. Researchers first used root sensitivity measures with transfer functions and then applied eigenvalue sensitivity measures with the state-space approach.

However, none of the early work used the resulting sensitivity measures to analytically find a globally least sensitive structure. It was only in 1984 that Tavşanoğlu and Thiele [169] published "the first purely analytical attempt to synthesize minimum sensitivity state-space realizations for linear systems" [62]. I first discuss the measure they minimized (following Gevers and Li, I will call it $M_{L_{12}}$) in Section 4.1. I also include Lutz and Hakimi's extension to MIMO systems [109]. Section 4.2 describes Gevers and Li's M_{L_2} measure which refines the $M_{L_{12}}$ metric and also lays some groundwork for their synthetic measure. The section also includes Xiao's extension [191] of M_{L_2} to account for 0 and ± 1 entries. Section 4.3 develops results on root sensitivity of polynomials and then goes on to describe pole-zero sensitivity measures that incorporate the state-space approach and eigenvalue sensitivity. Section 4.4 describes how to frequency weight each of the previous measures. This frequency weighting can help to account for colored inputs or to emphasize more important frequency bands and deemphasize the ones where minimum FWL sensitivity is not important.

DSP - Stochastic Measures

Section 4.5 switches to the stochastic realm and describes the ubiquitous roundoff noise gain measure. Section 4.6 develops the stochastic model of coefficient quantization errors. Section 4.7 describes Gevers and Li's new synthetic measure, which combines the stochastic coefficient quantization error and roundoff noise gain measures into one.

Controls - Deterministic Measures

Section 4.8 switches to the controls context and describes the differences between the DSP measures and controller measures, reiterating some of the ideas and warnings of Chapter 3. Section 4.9 includes the measures of sensitivity of the *entire* closed loop to perturbations in the controller coefficients. It includes Madievski *et al.*'s development [111] of an operator-based norm to measure controller sensitivity while taking into account the CT plant, not a DT approximation of it. Section 4.10 develops the modern controls paradigm and poses the sensitivitity and roundoff noise minimization problems in the framework of stability robustness.

Controls - Stochastic Measures

Section 4.11 covers the stochastic closed-loop roundoff noise gain metric. Finally, Section 4.12 describes some of the other measures that I did not have time to include. It contains references and a short description for each of the measures, and I leave it for future work to develop them in more detail.

4.1 The $M_{L_{12}}$ Measure

Coefficient quantization changes a system's transfer function but does not on its own introduce any non-linearity. Also, it only 'affects' the system once, when the coefficients of the realization are first rounded. Thus, a typical method to measure coefficient quantization's impact is to measure the sensitivity of the transfer function

Measure	Section	Source
DSP - Deterministic measures		
$M_{L_{12}}$ - SISO	4.1	[62,169,172,173]
$M_{L_{12}}$ - MIMO	4.1	[109]
M_{L_2}	4.2	[62,141,146,191,194]
M_{pz}	4.3	[62,157,184,185]
$M^*_{L_{12}}$	4.4	[62,173]
DSP - Stochastic Sources		
G	4.5	[62,78,79]
Stochastic coefficient quantization	4.6	[7,40,55,62,77,91,94,119,120]
G_T	4.7	[62]
Controls - Deterministic measures		
$M_{cl,L_{12}}$ - SISO	4.9	[62]
M_{cl,L_2} - SISO	4.9	[111]
Stability Robustness - μ_0, r_R	4.10	[100]
Controls - Stochastic measures		
Stability Robustness - μ_0, r_R	4.10	[54,55]
G_{cl}	4.11	[62]
$G_{T,cl}$	–	[62]
LQG cost - J	5.10	[62,105,119,120,185,186,196]

Table 4.1: Measures.

to coefficient perturbations, because the rounding process perturbs each coefficient by up to $\frac{2^{-B_c}}{2}$, where B_c is the number of bits for the fractional part of the coefficient wordlength. There are at least two different definitions of a SISO system's sensitivity. Both are developed with a state-space approach.

SISO Systems

Consider the standard discrete-time state-space description in (2.4) with a single input and a single output i.e. $B \in \mathbb{R}^{n\times 1}$, and $C \in \mathbb{R}^{1\times n}$ and define the following functions:

$$\begin{aligned}\frac{\partial H(z)}{\partial c_i} &= \frac{\partial}{\partial c_i}(C(zI-A)^{-1}B+D)\\ &= [(zI-A)^{-1}B]_i.\\ F(z) \triangleq \frac{\partial H(z)}{\partial C} &= (zI-A)^{-1}B \qquad (4.2)\end{aligned}$$

where $F(z) \in \mathbb{R}^{n \times 1}$. Call the impulse response sequence from the input to the ith state variable f_i. Similarly,

$$G(z) \triangleq \frac{\partial H(z)}{\partial B} = C(zI - A)^{-1} \tag{4.4}$$

where $G(z) \in \mathbb{R}^{1 \times n}$. Call the impulse response sequence from the ith state variable to the output g_i. Note the slight abuse of notation: $\frac{\partial}{\partial B}$, a column operator, results in a row vector, and $\frac{\partial}{\partial C}$, a row operator, results in a column vector. Finally,

$$\begin{aligned} \left[\frac{\partial H(z)}{\partial A}\right]_{ij} &= C(zI - A)^{-1} e_i e_j^T (zI - A)^{-1} B \\ &= G(z) e_i e_j^T F(z) \\ &= G_i(z) F_j(z) && (4.5) \\ \Rightarrow \frac{\partial H(z)}{\partial A} &= G^T(z) F^T(z) && (4.7) \end{aligned}$$

where e_i and e_j are unit vectors.

Now, to measure the total effect over all frequencies, Tavşanoğlu and Thiele [169, 172] proposed the following measure[1]

$$M_{L_{12}} \triangleq \left\| \frac{\partial H(z)}{\partial A} \right\|_1^2 + \left\| \frac{\partial H(z)}{\partial B} \right\|_2^2 + \left\| \frac{\partial H(z)}{\partial C} \right\|_2^2 \tag{4.8}$$

where $\|\cdot\|_p$, the L_p norm of a function $f(t) \in \mathbb{C}^{n \times m}$ is defined as

$$\|f\|_p = \left(\frac{1}{2\pi} \int_0^{2\pi} \|f(e^{j\omega})\|_F^p \, d\omega \right)^{1/p} \tag{4.9}$$

At first glance, a 1-norm seems a little out of place. The natural correspondence between the time and frequency domains with the 2-norm is energy, but the relationship of the 1-norm to the time domain is obscure. In fact, Thiele *et al.* [169] used it primarily for mathematical convenience. Gevers and Li [62] and Rao [146] develop a measure with a more natural 2-norm for the first term also (see Section 4.2).

[1] As in the case of most of the other measures, this one is being developed to be optimized over the space of similarity transformations. Thus, the D term is left out since it is coordinate independent.

Both Jackson [81] and Mullis and Roberts [151] use a similar sensitivity formula in their textbooks:

$$S_{ij}(z) = G_i(z)F_j(z) \tag{4.10}$$

Thus, $S_{ij} = \left[\dfrac{\partial H(z)}{\partial A}\right]_{ij}$.

I shall evaluate the second and third terms in (4.8) and then return to the slightly more complicated first term.

$$\begin{aligned}
\left\|\frac{\partial H(z)}{\partial B}\right\|_2^2 &= \frac{1}{2\pi}\int_0^{2\pi} \left\|\frac{\partial H(e^{j\omega})}{\partial B}\right\|_F^2 \, d\omega \\
&= \frac{1}{2\pi}\int_0^{2\pi} \sum_{i=1}^{n} \left|G_i(e^{j\omega})\right|^2 \, d\omega \\
&= \sum_{i=1}^{n} \frac{1}{2\pi}\int_0^{2\pi} G_i(e^{-j\omega})G_i(e^{j\omega}) \, d\omega \\
&= \sum_{i=1}^{n} \|g_i\|_2^2 \\
&= \operatorname{tr}(W_o)
\end{aligned}$$

where W_o is the observability Gramian of the system. Similarly,

$$\left\|\frac{\partial H(z)}{\partial B}\right\|_2^2 = \operatorname{tr}(W_c)$$

The energy in the impulse response sequence from the ith state variable to the output is the (i,i)th entry of W_o. Similarly the energy of the impulse response sequence from the input to the ith state variable is the (i,i)th entry of W_c.

For the first term, the Cauchy-Schwarz inequality gives an upper bound:

$$\begin{align}
\left\|\frac{\partial H(z)}{\partial A}\right\|_1^2 &= \left(\frac{1}{2\pi}\int_0^{2\pi} \|G^T(e^{j\omega})F^T(e^{j\omega})\|_F \, d\omega\right)^2 \notag \\
&= \left(\frac{1}{2\pi}\int_0^{2\pi} [G(e^{-j\omega})G^T(e^{j\omega})]^{1/2}[F^T(e^{-j\omega})F(e^{j\omega})]^{1/2} \, d\omega\right)^2 \notag \\
&\leq \left(\frac{1}{2\pi}\int_0^{2\pi} G(e^{-j\omega})G^T(e^{j\omega}) \, d\omega\right)\left(\frac{1}{2\pi}\int_0^{2\pi} F(e^{-j\omega})F^T(e^{j\omega}) \, d\omega\right) \tag{4.11} \\
&= \left\|\frac{\partial H(z)}{\partial B}\right\|_2^2 \left\|\frac{\partial H(z)}{\partial C}\right\|_2^2 \tag{4.12}
\end{align}$$

Thus, combining all three terms, we get the following upper bound

$$M_{L_{12}} \leq \bar{M}_{L_{12}} = \left\| \frac{\partial H(z)}{\partial B} \right\|_2^2 \left\| \frac{\partial H(z)}{\partial C} \right\|_2^2 + \left\| \frac{\partial H(z)}{\partial B} \right\|_2^2 + \left\| \frac{\partial H(z)}{\partial C}{}_2^2 \right\| \tag{4.13}$$
$$= \operatorname{tr}(W_c) \operatorname{tr}(W_o) + \operatorname{tr}(W_o) + \operatorname{tr}(W_c)$$

Now would be a good time to take a short detour and develop scaling in the state-space context. Since the (i, i)th entry of W_c indicates the energy in the impulse response sequence from the input to the ith state variable, l_2 scaling (see Section 2.3) should result in a W_c with all diagonal elements equal to unity (or $\frac{1}{\alpha^2}$, depending on the scaling factor α – see Section 2.3). Thus, we must first determine how W_c changes under a similarity transformation.

$$W_c = \sum_{i=0}^{\infty} (A^i B)(A^i B)^T$$

transforms to

$$\sum_{i=0}^{\infty} ((T^{-1}AT)^i T^{-1}B)(T^{-1}AT)^i T^{-1}B)^T$$
$$= \sum_{i=0}^{\infty} T^{-1}(A^i B)(A^i B)^T T^{-T}$$
$$= T^{-1} W_c T^{-T} \tag{4.14}$$

To get scaled diagonal elements, simply apply the diagonal scaling matrix $[T]_{ii} = \alpha\sqrt{[W_c]_{ii}}$. Then, $\operatorname{tr}(W_c) = \frac{n}{\alpha^2}$. That ends the detour.

Thus, for an l_2 scaled system, $\bar{M}_{L_{12}} = \frac{n}{\alpha^2}(\operatorname{tr}(W_o) + 1) + \operatorname{tr}(W_o)$.

A property of the Cauchy-Schwarz inequality is that equality holds in (4.11) if and only if $F(e^{-j\omega})F^T(e^{j\omega}) = \gamma G(e^{-j\omega})G^T(e^{j\omega})$ for some real constant γ. This condition immediately translates into $W_c = \gamma W_o$ for (4.12) to hold with equality [169].

I shall return to the upper bound (4.13) in Section 5.2 to discuss how its value changes under similarity transformations and to describe the set of optimal similarity transformations that minimize it. Coincidentally, in some cases, the transformation that minimizes the upper bound also happens to minimize the measure itself! Finally, this upper bound is also intimately tied to the roundoff noise gain measure (see Section

4.5). The functions trace and dgram in MATLAB can be used to easily compute the upper bound.

Next, I present the extension of the $M_{L_{12}}$ sensitivity measure to the MIMO case.

MIMO Systems

Note that the above definitions only hold true in the SISO case. Lutz and Hakimi [109] first proposed and solved the more complicated sensitivity function in the MIMO case, and the development here closely follows their's. The main extension required is the theory of derivatives of matrices with respect to matrices. In the sequel, $[\cdot]_{i\cdot}$ indicates the ith row of a matrix while $[\cdot]_{\cdot j}$ indicates the jth column of a matrix.

In the MIMO case, $B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}$. Let $F(z)$ and $G(z)$ be defined as before

$$F(z) = (zI - A)^{-1}B$$
$$G(z) = C(zI - A)^{-1}$$

where $F \in \mathbb{R}^{n\times m}$ and $G \in \mathbb{R}^{p\times n}$.

Then, using the definition of the matrix derivative in Appendix C.6, $\dfrac{\partial H(z)}{\partial B}$ will be a matrix of order $pn \times m^2$ where the ijth partition will be

$$\frac{\partial [H(z)]_{ij}}{\partial B} = G^T(z)E_{ij} = G^T(z)e_i e_j^T = \left[G^T(z)\right]_{\cdot i} e_j^T \tag{4.15}$$

where $E_{ij} = e_i e_j^T$ is a $p\times m$ elementary matrix. Note that the matrix in each partition has the same shape as B.

Applying the vec operator, we can rewrite (4.15) as

$$\begin{aligned}\frac{\partial H(z)}{\partial B} &= \left[(\text{vec } G^T)0\cdots 0 \,\vdots\, 0(\text{vec } G^T)0\cdots 0 \,\vdots\, \cdots \,\vdots\, 0\cdots 0(\text{vec } G^T) \right] \\ &= (\text{vec } G^T(z))(\text{vec } I_m)^T\end{aligned}$$

where each 0 matrix is of size $pn \times (m-1)$.

Similarly, $\dfrac{\partial H}{\partial C}$ will be of order $p^2 \times nm$:

$$\frac{\partial [H(z)]_{ij}}{\partial C} = E_{ij}F^T(z) = e_i(F(z)e_j)^T = e_i \left[F(z)\right]_{\cdot j}^T \tag{4.16}$$

where $E_{ij} = e_i e_j^T$ is a $p \times m$ elementary matrix.

Rewriting (4.16),

$$
\begin{aligned}
\frac{\partial H(z)}{\partial C} &= \left[\begin{array}{c}
(\text{vec } F)^T \\
0 \\
\vdots \\
0 \\
\hdashline
0 \\
(\text{vec } F)^T \\
\vdots \\
0 \\
\hdashline
\vdots \\
\hdashline
0 \\
\vdots \\
0 \\
(\text{vec } F)^T
\end{array}\right] \\
&= (\text{vec } I_p)(\text{vec } F(z))^T
\end{aligned}
$$

Finally, to derive $\dfrac{\partial H(z)}{\partial A}$ we use Graham's "First Transformation Principle" [143] (see appendix C.4) and the following property of matrices:

$$
\begin{aligned}
\frac{\partial M^{-1}}{\partial x} &= -M^{-1}\frac{\partial M}{\partial x} M^{-1} \\
\Rightarrow \frac{\partial H(z)}{\partial a_{ij}} &= C\frac{\partial (zI - A)^{-1}}{\partial a_{ij}} B \\
&= -C(zI - A)^{-1}\frac{\partial (zI - A)}{\partial a_{ij}} (zI - A)^{-1}B \\
&= G(z)E_{ij}F(z)
\end{aligned}
$$

where E_{ij} is of size $n \times n$; $\dfrac{\partial H(z)}{\partial a_{ij}}$ is of dimension $p \times m$. Applying the Transformation Principle,

$$
\frac{\partial [H(z)]_{ij}}{\partial A} = G^T(z)E_{ij}F^T(z)
$$

Thus $\frac{\partial[H(z)]_{ij}}{\partial A} = \left[G^T(z)\right]_{\cdot i}[F(z)]^T_{\cdot j}$ which allows us to write

$$\frac{\partial H(z)}{\partial A} = \left[\begin{array}{c:c:c:c} \left[G^T\right]_{\cdot 1}[F]^T_{\cdot 1} & \left[G^T\right]_{\cdot 1}[F]^T_{\cdot 2} & \cdots & \left[G^T\right]_{\cdot 1}[F]^T_{\cdot m} \\ \hdashline \left[G^T\right]_{\cdot 2}[F]^T_{\cdot 1} & \left[G^T\right]_{\cdot 2}[F]^T_{\cdot 2} & \cdots & \left[G^T\right]_{\cdot 2}[F]^T_{\cdot m} \\ \hdashline \vdots & \vdots & & \vdots \\ \hdashline \left[G^T\right]_{\cdot p}[F]^T_{\cdot 1} & \left[G^T\right]_{\cdot p}[F]^T_{\cdot 2} & \cdots & \left[G^T\right]_{\cdot p}[F]^T_{\cdot m} \end{array}\right]$$
$$= (\text{vec } (G^T))(\text{vec } F)^T$$

The dimensions of $\frac{\partial H(z)}{\partial A}$ are $pn \times mn$.

Next, we apply norms as in the SISO case to get a sensitivity measure. To evaluate each term in $M_{L_{12}}$ (4.8) with the new expressions for the derivatives, we first need to evaluate the Frobenius norm of each term.

$$\begin{aligned} \left\|\frac{\partial H}{\partial A}\right\|_F^2 &= \left[\text{vec } \frac{\partial H}{\partial A}\right]^T \left[\text{vec } \frac{\partial H}{\partial A}\right] \\ &= [\text{vec } ((\text{vec } G^T)(\text{vec } F)^T)]^T[\text{vec } ((\text{vec } G^T)(\text{vec } F)^T)] \\ &= [\text{vec } F \otimes \text{vec } G^T]^T[\text{vec } F \otimes \text{vec } G^T] \\ &= [\text{vec } F^T \otimes \text{vec } G][\text{vec } F \otimes \text{vec } G^T] \\ &= [\text{vec } F^T \text{vec } F] \otimes [\text{vec } G \text{vec } G^T] \\ &= \|F\|_F^2\|G\|_F^2 \end{aligned}$$

(See [66] for properties of the vec and $\otimes$ (Kronecker product) operators.)

Similarly, for

$$\begin{aligned} \left\|\frac{\partial H}{\partial B}\right\|_F^2 &= \left[\text{vec } \frac{\partial H}{\partial B}\right]^T \left[\text{vec } \frac{\partial H}{\partial B}\right] \\ &= [\text{vec } ((\text{vec } G^T)(\text{vec } I_m)^T)]^T[\text{vec } ((\text{vec } G^T)(\text{vec } I_m)^T)] \\ &= m\|G\|_F^2 \end{aligned}$$

and

$$\begin{aligned}\left\|\frac{\partial H}{\partial C}\right\|_F^2 &= \left[\text{vec}\ \frac{\partial H}{\partial C}\right]^T \left[\text{vec}\ \frac{\partial H}{\partial C}\right] \\ &= [\text{vec}\ ((\text{vec}\ I_p)(\text{vec}\ F)^T)]^T[\text{vec}\ ((\text{vec}\ I_p)(\text{vec}\ F)^T)] \\ &= p\|F\|_F^2\end{aligned}$$

Then, substitituting the above into (4.8),

$$\begin{aligned}M_{L_{12}} &= \left(\frac{1}{2\pi}\int_0^{2\pi}\left\|\frac{\partial H}{\partial A}\right\|_F d\omega\right)^2 + \frac{1}{2\pi}\int_0^{2\pi}\left\|\frac{\partial H}{\partial B}\right\|_F^2 d\omega + \frac{1}{2\pi}\int_0^{2\pi}\left\|\frac{\partial H}{\partial C}\right\|_F^2 d\omega \\ &= \left(\frac{1}{2\pi}\int_0^{2\pi}\|F\|_F\,\|G\|_F\ d\omega\right)^2 + \frac{1}{2\pi}\int_0^{2\pi} m\,\|G\|_F^2\ d\omega + \frac{1}{2\pi}\int_0^{2\pi} p\,\|F\|_F^2\ d\omega\end{aligned}$$

Using the Cauchy Schwarz inequality on the first term, as in the SISO case, one gets the bound

$$M_{L_{12}} \leq \bar{M}_{L_{12}} = \text{tr}\,(W_c)\,\text{tr}\,(W_o) + m\,\text{tr}\,(W_o) + p\,\text{tr}\,W_c \tag{4.17}$$

I will present the transformation that minimizes the bound in Section 5.2.

4.2 The M_{L_2} Sensitivity Measure

Noting that the $M_{L_{12}}$ measure uses a 1-norm on the $\dfrac{\partial H(z)}{\partial A}$ term mostly for mathematical convenience, [62] instead replaces it with a more natural 2-norm, resulting in

$$M_{L_2} \triangleq \left\|\frac{\partial H(z)}{\partial A}\right\|_2^2 + \left\|\frac{\partial H(z)}{\partial B}\right\|_2^2 + \left\|\frac{\partial H(z)}{\partial C}\right\|_2^2$$

(We return to considering SISO systems.) Rao [146] developed the M_{L_2} measure also, but I chose to include Gevers' and Li's development since it is easier to follow. Furthermore, they also present the solution to finding a structure that minimizes this measure (see Section 5.3), while Rao [146] presents sub-optimal solutions that have low sensitivity, but he does not find the optimal one.

The second and third terms are the same as before. The new first term is

$$\begin{aligned}\left\|\frac{\partial H(z)}{\partial A}\right\|_2^2 &= \frac{1}{2\pi}\int_0^{2\pi}\left\|\frac{\partial H(e^{j\omega})}{\partial A}\right\|_F^2 \, d\omega \\ &= \frac{1}{2\pi}\int_0^{2\pi}\operatorname{tr}\left\{\frac{\partial H(e^{j\omega})}{\partial A}\left(\frac{\partial H(e^{j\omega})}{\partial A}\right)^H\right\} d\omega \\ &= \operatorname{tr}\left\{\frac{1}{2\pi}\int_0^{2\pi}\left(\frac{\partial H(e^{j\omega})}{\partial A}\right)\left(\frac{\partial H(e^{j\omega})}{\partial A}\right)^H d\omega\right\}\end{aligned}$$

Noting that $\frac{\partial H(z)}{\partial A} = G^T(z)F^T(z)$ and that $(zI-A)^{-1} = \sum_{i=0}^{\infty} A^i z^{-(i+1)}$,

$$\begin{aligned}\frac{\partial H(z)}{\partial A} &= G^T(z)F^T(z) \\ &= (C(zI-A)^{-1})^T((zI-A)^{-1}B)^T \\ &= \left(\sum_{i=0}^{\infty} CA^i z^{-(i+1)}\right)^T\left(\sum_{i=0}^{\infty} A^i B z^{-(i+1)}\right)^T \\ &= \sum_{i=0}^{\infty} g^T(i) z^{-(i+1)} \sum_{j=0}^{\infty} f^T(j) z^{-(j+1)}\end{aligned}$$

where $g(i) = CA^i$ and $f(j) = A^j B$. Then, combining the two summations,

$$\begin{aligned}\frac{\partial H(z)}{\partial A} &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} g^T(i) f^T(j) z^{-(i+j+2)} \\ &= \sum_{k=0}^{\infty} h(k) z^{-(k+2)} \qquad (4.18)\end{aligned}$$

where $h(k)$ is the $n \times n$ matrix $g^T(i)f^T(j)$ for $i+j=k$. Finally, we can write

$$\begin{aligned}\left\|\frac{\partial H(z)}{\partial A}\right\|_2^2 &= \operatorname{tr}\left\{\frac{1}{2\pi}\int_0^{2\pi}\left(\frac{\partial H(e^{-j\omega})}{\partial A}\right)^T \frac{\partial H(e^{j\omega})}{\partial A} \, d\omega\right\} \\ &= \operatorname{tr}\left\{\frac{1}{2\pi}\int_0^{2\pi}\left(\sum_{k=0}^{\infty} h^T(k) e^{-jk\omega} e^{-j2\omega}\right)\left(\sum_{k=0}^{\infty} h(k) e^{jk\omega} e^{j2\omega}\right) d\omega\right\} \qquad (4.19)\end{aligned}$$

Since $\frac{1}{2\pi}\int_0^{2\pi} e^{jk\omega} \, d\omega = 0$ if $k \neq 0$, (4.19) reduces to

$$\begin{aligned}\left\|\frac{\partial H(z)}{\partial A}\right\|_2^2 &= \operatorname{tr}\sum_{k=0}^{\infty} h(k)^T h(k) \frac{1}{2\pi}\int_0^{2\pi} d\omega \\ &= \operatorname{tr}\sum_{k=0}^{\infty} h(k)^T h(k) \\ &\triangleq \operatorname{tr}(W_A)\end{aligned}$$

resulting in the measure

$$M_{L_2} = \mathrm{tr}\,(W_A) + \mathrm{tr}\,(W_o) + \mathrm{tr}\,(W_c)$$

Note that Rao's alternative expressions for M_{L_2} is more useful for computational purposes:

$$\begin{aligned} M_{L_2} &= \mathrm{tr}\,(W_c) + \mathrm{tr}\,(W_o) + \mathrm{tr}\,(W_c)\mathrm{tr}\,(W_o) + 2\sum_{i=0}^{\infty}\sum_{j=1}^{n}\left[W_o^T\right]_{j\cdot} A^i e_j \sum_{k=1}^{n} e_k^T A^i \left[W_c\right]_{\cdot k} \\ &= \mathrm{tr}\,(W_c) + \mathrm{tr}\,(W_o) - \mathrm{tr}\,(W_c)\mathrm{tr}\,(W_o) + 2x^T M y \end{aligned}$$

where x and y are vectors defined as

$$\begin{aligned} x_i &= x_{p_i}^T W_o y_{p_i} \\ y_i &= y_{p_i}^T W_c x_{p_i} \\ M &= \begin{bmatrix} \frac{1}{1-|\lambda_1|^2} & \frac{1}{1-\lambda_1\lambda_2^H} & \cdots & \frac{1}{1-\lambda_1\lambda_n^H} \\ \frac{1}{1-\lambda_2\lambda_1^H} & \frac{1}{1-|\lambda_2|^2} & \cdots & \frac{1}{1-\lambda_2\lambda_n^H} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{1-\lambda_n\lambda_1^H} & \cdots & \frac{1}{1-\lambda_n\lambda_2^H} & \frac{1}{1-|\lambda_n|^2} \end{bmatrix} \end{aligned}$$

x_{p_i} and y_{p_i} are the right and left eigenvectors respectively of A. See [146] for the derivation.

Yan and Moore [194] also give the measure in relation to the solution of a Lyapunov equation.

The M_{L_2} minimizing SISO realization is in Section 5.3.

Madievski *et al.* [111] develop an operator-based MIMO version of the M_{L_2} measure for minimizing sensitivity in the sampled-data controls setup. One should be able to easily apply it in the simpler filtering case also (see Section 4.9).

Relationship between M_{L_2} and $M_{L_{12}}$ norms

Gevers and Li [62, Chapter 5] prove that $M_{L_2} \geq M_{L_{12}}$, with equality only in the pathological case that $\left\|\frac{\partial H(z)}{\partial A}\right\|_F$ is constant over all ω. Furthermore, they investigate how a realization that minimizes the $M_{L_{12}}$ measure performs under M_{L_2}. Their example shows that an $M_{L_{12}}$ optimal realization performs well under M_{L_2} and vice versa.

Extending the M_{L_2} measure to account for 0, ±1 terms

Noting that quantization does not perturb 0 and ±1 terms, Xiao [191] suggests a modified M_{L_2} metric for SISO systems. In fact, *any integer coefficient will be realized exactly and will not contribute to sensitivity or roundoff error.* Also, any coefficients that have a fractional part that is a sum of negative powers of two will be realized exactly as long as the wordlength is long enough. However, these coefficients *will* contribute to roundoff error. Xiao's measure does not account for these situations. (It is a trivial modification to include the integer coefficients, and I will point it out below).

Xiao's modified measure, which he calls S_I, requires some additional terms. Let

$$
\begin{aligned}
S_{a_{ik}} &= \frac{\partial H(z)}{\partial a_{ik}} &= G(z)e_i e_k^T F(z)\phi_{ik} \\
S_{b_i} &= \frac{\partial H(z)}{\partial b_i} &= G(z)e_i\varphi_i \\
S_{c_k} &= \frac{\partial H(z)}{\partial c_k} &= e_k^T F(z)\psi_k
\end{aligned}
$$

where

$$
\phi_{ik} = \begin{cases} 0 & \text{for } a_{ik} = 0, \pm 1 \\ 1 & \text{for } a_{ik} \neq 0, \pm 1 \end{cases}
$$

$$
\varphi_i = \begin{cases} 0 & \text{for } b_i = 0, \pm 1 \\ 1 & \text{for } b_i \neq 0, \pm 1 \end{cases}
$$

$$
\psi_k = \begin{cases} 0 & \text{for } c_k = 0, \pm 1 \\ 1 & \text{for } c_k \neq 0, \pm 1 \end{cases}
$$

The simple modification to exclude integer coefficients is to let $\phi_{ik}, \varphi_i,$ and ψ_k be zero if the corresponding coefficient is integral. Then,

$$
S_I \triangleq \sum_{i=1}^{n}\sum_{k=1}^{n}\phi_{ik}\,[\,c\ 0\,]\,R(i,k)\,[\,c\ 0\,]^T + \sum_{i=1}^{n}\varphi_i[W_o]_{ii} + \sum_{k=1}^{n}\psi_k[W_c]_{kk}
$$

where $R(i,k)$ is a symmetric matrix that satisfies the equation

$$R(i,k) - \begin{bmatrix} A & e_i e_k^T \\ 0 & A \end{bmatrix} R(i,k) \begin{bmatrix} A^T & 0 \\ e_k e_i^T & A^T \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix} \begin{bmatrix} 0 \\ b \end{bmatrix}^T$$

For the proof, refer to [191].

For a fully parametrized system, computing S_I requires solving n^2 Lyapunov equations. Moreover, Xiao does not propose how to find the realization that minimizes S_I. Sparse realizations generated by Amit and Shaked's $0,1$ algorithm (see Section 5.11) are good candidates for comparing M_{L_2}, S_I, the corresponding roundoff noise gain measure G, and the measure used by Amit and Shaked.

4.3 Pole-Zero Sensitivity

Mathematicians have studied root sensitivity of general polynomials for a long time[2]. The results used in linear system theory are largely re-interpretations of older results. Kaiser's analysis [87,88] was one of the first to consider how pole sensitivity changes with sampling rate and filter order. He gave word length bounds, also as functions of sampling rate and filter order. Not only is following Kaiser's development of root sensitivity instructive, but it motivates study of the FWL problem and the search for low sensitivity system implementations.

Polynomial Root Sensitivity

Consider a CT TF and the bilinear discretization of its denominator[3],$D(z^{-1})$, defined as

$$\begin{aligned} D(z^{-1}) &= \prod_{i=1}^{n} (s - p_i)\Big|_{s \to \frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}} \\ &= \prod_{i=1}^{n} \left(1 - \frac{(1 + \frac{p_i T}{2})}{(1 - \frac{p_i T}{2})} z^{-1} \right) \end{aligned}$$

[2] In mathematics, a long time signifies a century or more.

[3] D is not related to the system matrix D in state-space realizations but rather stands for the denominator of the TF.

Thus, the pole p_i gets mapped to

$$z_i = \frac{1 - \frac{p_i T}{2}}{1 + \frac{p_i T}{2}}$$

See Appendix C.3 for a more detailed computation. Now, if $|p_i T| \ll 1$,

$$\frac{1}{1 + \frac{p_i T}{2}} \approx 1 - \frac{p_i T}{2}$$
$$\Rightarrow z_i \approx 1 - p_i T$$

One can see that as T gets smaller, all the DT poles will cluster around $z = 1$.

To estimate the order of the perturbations necessary to move a root of $D(z^{-1})$ to $z^{-1} = 1$, consider:

$$D(z^{-1})|_{z^{-1}=1} = \prod_{i=1}^{n}(1 - z_i) = \prod_{i=1}^{n} p_i T \tag{4.20}$$

Another form of the denominator is

$$D(z^{-1})|_{z^{-1}=1} = 1 + \sum_{i=1}^{n} a_i z^{-i}|_{z^{-1}=1} = 1 + \sum_{i=1}^{n} a_i \tag{4.21}$$

If any of the a_i are changed by $F_0 \triangleq 1 + \sum_{j=1}^{n} a_j$, then (4.21) can be zero, which would mean a pole at $z = 1$. Though this bound is very crude, one can still comment on the relationship of wordlength and sampling rate/system order.

Equating (4.20) and (4.21) implies that coefficient accuracy is affected by the sampling rate and by the system order. Thus, going from an nth order filter to a $(2n)$th order filter will require approximately twice as many digits of accuracy to represent the a_i. Similarly, doubling the sampling rate for an nth order filter will require n additional bits to represent the a_i. In both cases, the crudest perturbation bound for destabilizing the transfer function will significantly decrease, thus requiring additional wordlength.

Tightening the bound further, Kaiser computes the sensitivity of the poles to small changes in the coefficients. Equating the two forms of $D(z^{-1})$,

$$1 + \sum_{l=1}^{n} a_l z^{-l} = \prod_{j=1}^{n}(1 - \frac{z^{-1}}{z_j})$$

leads to the sensitivity with respect to a coefficient:

$$\frac{\partial z_i}{\partial a_k} = \frac{z_i^{k+1}}{\prod\limits_{\substack{l=1 \\ l \neq i}}^{n} \left(1 - \frac{z_i}{z_l}\right)}$$

The total differential change (to a first order approximation) is

$$dz_i = \sum_{k=1}^{n} \frac{\partial z_i}{\partial a_k} \; da_k$$

Note that as the poles cluster together, the term $1 - \frac{z_i}{z_l}$ will get smaller and its reciprocal will get larger, displaying the commonly known fact that pole sensitivity increases as the poles get closer together.

With state-space structures, the eigenvalues of the A matrix are the roots of the denominator and determine stability. Thus, I turn to eigenvalue analysis next.

Eigenvalue Sensitivity

One can measure the sensitivity of the poles, which are the eigenvalues of A, and of the zeros, the eigenvalues of $Z = A - BD^{-1}C$ (assuming a SISO system and a non-zero D term). Skelton and Wagie [157] considered pole sensitivity minimization and defined pole sensitivity as

$$\Psi_p = \sum_{i=1}^{n} \Psi_{pi}$$

$$\text{where } \Psi_{pi} = \left\| \frac{\partial \lambda_i}{\partial A} \right\|_F^2 \qquad (4.22)$$

Williamson [184] developed a norm to measure the sensitivity of zeros. He defines

$$\Psi_z = \sum_{i=1}^{n} \Psi_{zi}$$

$$\text{where } \Psi_{zi} = \left\| \frac{\partial v_i}{\partial Z} \right\|_F^2$$

where v_i is the ith zero. Two observations about Ψ_z are in order:

(i) This measure only applies to systems with a non-zero D matrix, since Z is well-defined only in those cases.

(ii) Gevers and Li [62] point out that this metric measures the sensitivity of the zeros to the entries of the Z matrix, *not* to the entries of the coefficient matrices (A, B, C, D). They present a much more general measure that combines both pole and zero sensitivity:

$$M_{pz} \triangleq \sum_{i=1}^{n} w_{\lambda_i} \left\| \frac{\partial \lambda_i}{\partial A} \right\|_F^2 + w_{v_i} \left(\left\| \frac{\partial v_i}{\partial A} \right\|_F^2 + \left\| \frac{\partial v_i}{\partial B} \right\|_F^2 + \left\| \frac{\partial v_i}{\partial C} \right\|_F^2 + \left\| \frac{\partial v_i}{\partial D} \right\|_F^2 \right)$$
$$= \sum_{i=1}^{n} w_{\lambda_i} \Psi_{pi} + w_{v_i} \Psi_{zi}$$

where the $\{w_{\lambda_i} \geq 0, i = 1, 2, ..., n\}$ and $\{w_{v_i} \geq 0, i = 1, 2, ..., n\}$ weight each term and reflect the importance the designer wants to place on the ith pole or zero.

Why Frobenius norms and not some other norm? Consider modeling each coefficient quantization error as a uniformly distributed zero mean random variable with variance $\frac{2^{-2B_c}}{12}$. Then, to a first-order approximation, the variances of the pole and zero deviations, $|\delta\lambda_i|$ and $|\delta v_i|$, are $\frac{2^{-2B_c}}{12}\Psi_{pi}$ and $\frac{2^{-2B_c}}{12}\Psi_{zi}$ [100]. I shall develop the stochastic coefficient quantization error model in Section 4.6.

Setting all the w_{λ_i} to 1 and the w_{v_i} to 0 in M_{pz} recovers Skelton and Wagie's measure.

4.3.1 Pole Sensitivity Minimization

Mathematicians have applied elaborate theory to eigenvalue sensitivity, or eigenvalue perturbation analysis, documenting their results in an extensive literature (see for example [163] and the references therein). The strongest result for eigenvalue sensitivity uses the theory of Gerschgorin disks [163]. The eigenvalue senstivity analyses that most authors (in controls and signal processing) use rest on the differentiability of λ with respect to A, which in turn requires distinct eigenvalues [62].

A derivation and discussion of eigenvalue sensitivity is listed in [62, chapter 6]. This derivation assumes distinct eigenvalues, i.e. algebraic multiplicity of λ_i is one for all i. Thus, A has a linearly independent set of eigenvectors. The result of the derivation is

$$\left(\frac{\partial \lambda_i}{\partial A}\right)^T = XE_iX^{-1} \tag{4.23}$$

where X is the matrix of right eigenvectors of A, and $E_i = e_ie_i^T$. Now we must compute the norm, $\Psi_{pi} = \left\|\frac{\partial \lambda_i}{\partial A}\right\|_F^2 = \text{tr}\left\{\left(\frac{\partial \lambda_i}{\partial A}\right)^T \frac{\partial \lambda_i}{\partial A}\right\}$

Independent eigenvectors allow the A matrix to be decomposed as $A = X\Lambda X^{-1}$ where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$. It immediately follows that $X^{-1}A = \Lambda X^{-1}$ which means that the rows of X^{-1} are the left eigenvectors of A. Call the matrix of left eigenvectors (arranged in columns) Y. Then, $Y^H = X^{-1}$ and $y_i^H x_i = 1$ (since $Y^H X = I$). Normalizing the right eigenvectors ($x_i^H x_i = 1$) gives

$$\begin{aligned} \left\|\frac{\partial \lambda_i}{\partial A}\right\|_F^2 &= \text{tr}\left\{\left(Xe_ie_i^TX^{-1}\right)\left(X^{-H}e_ie_i^TX^H\right)\right\} \\ &= \text{tr}\left(x_iy_i^Hy_ix_i^H\right) \\ &= \text{tr}\left(x_i^Hx_iy_i^Hy_i\right) \\ &= \|y_i\|_2^2 \end{aligned} \tag{4.24}$$

The sensitivity measure of an eigenvalue is always greater than or equal to one [62, page 137]. Then, the measure is minimized if all eigenvalue sensitivities equal one, and the minimum of the measure will be n.

From linear algebra [97, page 176], the following conditions are equivalent:

(i) A is normal

(ii) $AA^H = A^HA$

(iii) $X^HX = I$ where X is the matrix of normalized right eigenvectors

(iv) A has an orthogonal set of eigenvectors

Gevers and Li [62] prove the following

$$\|y_k\|_2^2 = 1 \Leftrightarrow x_k = y_k \Leftrightarrow X^HX = I$$

Thus, a normal matrix minimizes overall pole sensitivity. Even for the weighted pole sensitivity metric, a normal matrix still minimizes the measure since any non-normal realization will only increase (at least) one of the Ψ_{pi} (and thus raise the overall sensitivity).

Can similarity transformations change any matrix to normal form? Yes, assuming that it has a linearly independent set of eigenvectors, which is always true when all eigenvalues are distinct.

To transform an arbitrary A (with distinct eigenvalues) to normal form, apply the following similarity transformation

$$T = (XD^{-2}X^{H})^{1/2}Q$$

where X is the matrix of the right eigenvectors of A, D is any positive definite diagonal matrix, and Q is an arbitrary orthogonal matrix [62]. In the simplest case, let $D = Q = I$. Then, $T = (XX^{H})^{1/2}$. The square root always exists since a positive (semi) definite D guarantees that XDX^{H} will be positive (semi) definite (see [62] for a proof).

Several authors have used normality to measure how good a realization is. As auxiliary measures, Williamson [184] suggests

$$S_p = \left\|AA^T - A^TA\right\|_F^2; \qquad S_z = \left\|ZZ^T - Z^TZ\right\|_F^2$$

for pole sensitivity and zero sensitivity (section 4.3.2), respectively. Skelton and Wagie [157] also add a penalty in the design process based on $\left\|AA^T - A^TA\right\|_F^2$.

Most authors stop here. However, it is intriguing to probe a little further into all the examples that do not satisfy the assumptions required by the above development.

The astute reader may note that for normality, we can in fact drop the requirement that A have distinct eigenvalues. Instead, we can impose the milder condition that all eigenvalues of A have geometric multiplicity equal to one or equivalently, they have an index equal to one. This condition, also equivalent to each Jordan block having size 1×1, guarantees a linearly independent set of eigenvectors. However, we must be careful to not immediately take this result backwards and assume that eigenvalue sensitivity (when algebraic multiplicity is greater than one but geometric multiplicity is equal to one) is still minimized by the normality condition. A derivation for eigenvalues with algebraic multiplicity greater than one requires a higher order expansion of the perturbation and needs to be examined more carefully. Refer to [97,163] for more details on higher-order expansions.

Some authors have erroneously concluded that repeated eigenvalues have infinite sensitivity. That is not true since the sensitivity derivation they rely on assumes the differentiability of λ with respect to A which breaks down in the case of repeated eigenvalues. This sensitivity measure is undefined in those cases [62].

How is this eigenvalue sensitivity measure related to the one Kaiser develops for the poles of a TF? Kaiser uses a TF description and $\dfrac{\partial \lambda_i}{\partial a_k}$ while the state-space approach uses $\dfrac{\partial \lambda_i}{\partial a_{ij}}$. Thus, to relate the measures, one must relate a_k to a_{ij}. In general, the a_k will be related to several coefficients of A and their complex relationship can be deduced by the expansion of $\det(zI - A)$. It would also be interesting to characterize the relationship between the two sensitivities in terms of the coefficients and perhaps show an equivalence.

Finally, Kaiser sought to tightly bound the size of the minimum perturbation in an a_i that would destabilize the TF. In state-space, Gevers and Li [100,101] and Fialho and Georgiou [54] ask what is the minimum perturbation in an a_{ij} that would destabilize the system (see Section 4.10). The latter problem is easily extensible to search the space of similar realizations.

4.3.2 Zero Sensitivity Minimization

In the SISO case, zeros are the eigenvalues of $Z \triangleq A - BD^{-1}C$. Again, to simplify analysis, most authors assume that Z has distinct eigenvalues. Then, to derive the partial derivatives with respect to the coefficient matrices, Gevers and Li [62] apply the chain rule and use the earlier expressions due to Williamson [184]. In the following, $D = d$. D is used in the expressions that carry over to the MIMO case without modification while d is used in expressions specific to the SISO case.

For the A matrix,

$$\begin{aligned}\frac{\partial v_k}{\partial a_{ij}} &= \sum_{l,p} \frac{\partial v_k}{\partial z_{lp}} \frac{\partial z_{lp}}{\partial a_{ij}} = \frac{\partial v_k}{\partial z_{ij}} \\ \frac{\partial v_k}{\partial A} &= \frac{\partial v_i}{\partial Z} \end{aligned} \tag{4.25}$$

For B and C,

$$\frac{\partial v_k}{\partial c_j} = \sum_{l,p} \frac{\partial v_k}{\partial z_{lp}} \frac{\partial z_{lp}}{\partial c_j}$$

$$\frac{\partial z_{lp}}{\partial c_j} = \frac{\partial}{\partial c_j} [A - BD^{-1}C]_{lp} = \frac{\partial}{\partial c_j} [BD^{-1}C]_{lp} = \frac{-b_l}{d} \delta_{pj}$$

which results in

$$\begin{aligned}\frac{\partial v_k}{\partial c_j} &= \sum_{l=1}^{n} \frac{\partial v_k}{\partial z_{lj}} \frac{-b_l}{d} = -(BD^{-1})^T \left[\frac{\partial v_k}{\partial Z}\right]_{\cdot j} \\ \Rightarrow \frac{\partial v_k}{\partial C} &= -D^{-1}B^T \frac{\partial v_k}{\partial Z} \end{aligned} \tag{4.26}$$

Similarly,

$$\begin{aligned}\frac{\partial v_k}{\partial b_i} &= \sum_{p=1}^{n} \frac{\partial v_k}{\partial z_{ip}} \frac{-c_p}{d} = -D^{-1} \left[\frac{\partial v_k}{\partial Z}\right]_{i\cdot} C^T \\ \Rightarrow \frac{\partial v_k}{\partial B} &= -\frac{\partial v_k}{\partial Z} (d^{-1}) C^T \end{aligned} \tag{4.27}$$

Finally,

$$\begin{aligned}\frac{\partial v_k}{\partial d} &= \sum_{l,p} \frac{\partial v_k}{\partial z_{lp}} \frac{\partial z_{lp}}{\partial d} = \sum_{l,p} \frac{\partial v_k}{\partial z_{lp}} (-d^{-2} b_l c_p) \\ &= -d^{-2} \sum_{l} b_l \sum_{p} \frac{\partial v_k}{\partial z_{lp}} c_p = -d^{-2} \sum_{l} b_l \left[\frac{\partial v_k}{\partial Z}\right]_{l\cdot} C^T \\ &= -d^{-2} B^T \frac{\partial v_k}{\partial Z} C^T \end{aligned} \tag{4.28}$$

Now, to compute the norms, define

$$X_z \triangleq [x_{z_1}, ..., x_{z_n}]$$

$$Y_z \triangleq X_z^{-H}$$

to be the set of right and left eigenvectors of Z respectively. Then, substituting (4.25)-(4.28) into (4.23) and (4.24) and replacing λ_i with v_i and A with Z,

$$\begin{aligned}\Psi_{zi} &= \mathrm{tr}\,\{(y_{z_i}x_{z_i}^H)(y_{z_i}x_{z_i}^H)^H\} + \mathrm{tr}\,(\alpha_i^2 y_{z_i} y_{z_i}^H) \\ &\quad + \mathrm{tr}\,(\beta_i^2 x_{z_i} x_{z_i}^H) + \alpha_i^2\beta_i^2\end{aligned}$$

where

$$\alpha_i^2 \triangleq \left|d^{-1}x_{z_i}^H C^T\right|^2 = \left|d^{-1}Cx_{z_i}\right|^2 \tag{4.29}$$

$$\beta_i^2 \triangleq \left|d^{-1}B^T y_{z_i}\right|^2 \tag{4.30}$$

With normalized right eigenvectors $\{x_z\}$, Ψ_{zk} reduces to

$$\Psi_{zi} = y_i^H y_i(1+\alpha_i^2) + \beta_i^2 + \alpha_i^2\beta_i^2$$

Note that this measure is restricted to a narrow set of systems, those with a direct feedthrough term (equivalently, those with as many zeros as poles) *and* a Z matrix with a full set of distinct eigenvalues i.e. all zeros are distinct. Ideally, one would like results to the most general problem: sensitivity of the *transmission zeros* of a MIMO system to coefficients in the system matrices.

This problem seems to have attracted little interest relative to the pole sensitivity problem (presumably since the latter directly impacts stability), and would probably be a good research problem.

4.3.3 Combined Pole-Zero Sensitivity Minimization

Combining the results of the previous two sections, [62] express the M_{pz} measure as

$$M_{pz} = \sum_{i=1}^{2n} \mathrm{tr}\,(H_i H_i^H) + \mathrm{tr}\,(M_y) + \mathrm{tr}\,(M_x) + c$$

where

$$
\begin{aligned}
H_i &= w_{\lambda_i}^{1/2} y_{p_i} x_{p_i}^H, i = 1, ..., n \\
&= w_{v_i}^{1/2} y_{z_{i-n}} x_{z_{i-n}}^H, i = n+1, ..., 2n \\
M_y &= \sum_{i=1}^{n} w_{v_i} \alpha_i^2 y_{z_i} y_{z_i}^H \\
&= Y_z diag(w_{v_1} \alpha_1^2, ..., w_{v_n} \alpha_n^2) Y_z^H \\
M_x &= \sum_{k=1}^{n} w_{v_i} \beta_i^2 x_{z_i} x_{z_i}^H \\
&= X_z diag(w_{v_1} \beta_1^2, ..., w_{v_n} \beta_n^2) X_z^H \\
c &= \sum_{k=1}^{n} w_{v_i} \alpha_i^2 \beta_i^2
\end{aligned}
$$

The set of transformations that minimizes this combined measure appears in Section 5.4.

4.4 Frequency Weighted Sensitivity Minimization

With each of the previous sensitivity criteria, $M_{L_{12}}$, M_{L_2}, and M_{pz}, one can introduce a frequency dependent weighting term. FWL sensitivity may not be important at frequency bands where the system will never operate. Thus, one can sacrifice quality in some frequency ranges to improve the response in others.

Thiele [173] first examined frequency weighted sensitivity functions in the context of colored noise input. For example, if the input to the filter had a spectrum $\Psi(z)$, then the power in the sequence from the input to x_i, the ith state variable, will be $\frac{1}{2\pi}\int_0^{2\pi} F_i(e^{j\omega})F_i(e^{-j\omega})^T \left|\Psi(e^{j\omega})\right|^2 \, d\omega$. One would use this information to modify the scaling rule. Thiele solved the general problem of computing weighted Gramians which arise when using these modified sensitivity functions (see [173] for a discussion and solution of weighted Gramians). This solution addresses the important case of colored input as well as colored roundoff noise (which would result in a weighted observability Gramian). However, it leaves out the general case.

Gevers and Li [62] develop the more general framework. Let $W_A(z)$, $W_B(z)$, and $W_C(z)$ be scalar weighting functions (rational in z[4]) for the matrices A, B, and C respectively[5]. Then, the $M_{L_{12}}$ sensitivity function (4.8) changes to

$$M^*_{L_{12}} = \left\| W_A(z)\frac{\partial H(z)}{\partial A} \right\|_1^2 + \left\| W_B(z)\frac{\partial H(z)}{\partial B} \right\|_2^2 + \left\| W_C(z)\frac{\partial H(z)}{\partial C} \right\|_2^2$$

Factor $W_A(z)$ into $W_1(z)W_2(z)$ where $W_1(z)$ and $W_2(z)$ are any factors of $W_A(z)$. Then, using the Cauchy Schwarz inequality on the first term as before,

$$\begin{aligned} \left\| W_A(z)\frac{\partial H(z)}{\partial A} \right\|_1^2 &= \left\| W_1(z)G^T(z)W_2(z)F^T(z) \right\|_1^2 \\ &\leq \left\| W_1(z)G(z) \right\|_2^2 \left\| W_2(z)F(z) \right\|_2^2 \end{aligned}$$

The upper bound of the frequency weighted measure is

$$\begin{aligned} M^*_{L_{12}} \leq \bar{M}^*_{L_{12}} \triangleq &\left\| W_1(z)G(z) \right\|_2^2 \left\| W_2(z)F(z) \right\|_2^2 \\ &+ \left\| W_B(z)G(z) \right\|_2^2 + \left\| W_C(z)F(z) \right\|_2^2 \end{aligned}$$

Each of these terms can be thought of as a weighted Gramian. Call these terms $K_{o1}, K_{c2}, K_{oB}, \text{and} K_{cC}$ (the first term is the observability Gramian weighted by $W_1(z)$ and so on). Then, $\bar{M}^*_{L_{12}}$ reduces to

$$\bar{M}^*_{L_{12}} = \text{tr}\,(K_{o1})\text{tr}\,(K_{c2}) + \text{tr}\,(K_{oB}) + \text{tr}\,(K_{cC})$$

One can compute the value of $\bar{M}^*_{L_{12}}$ using the algorithm given in [173]. The transformation that minimizes this upper bound is given in Section 5.5.

The general idea of frequency weighting applies in many contexts and can be applied to most measures. The norm-based measures discussed so far can all naturally incorporate frequency weighting just as in the $M_{L_{12}}$ case. However, a method to compute the measure may not always follow so easily. For the $M^*_{L_{12}}$ case, Thiele developed the weighted Gramian solutions. The M_{L_2} and M_{pz} measures would require

[4] The condition of rational weighting functions is only required for a tractable computational routine.

[5] W_A here has nothing to do with the W_A term in the definition of M_{L_2} sensitivity.

something similar. Moreover, after developing the measure, the more interesting question is how to find the transformation that minimizes it?

Other measures, not based on frequency norms, may also include frequency dependent error tolerances. Crochiere [40], in his problem description for digital filter design, allows a frequency dependent error tolerance for the transfer function magnitude.

4.5 Roundoff Noise Gain

Roundoff noise gain, one of the earliest stochastic measures, originates from modeling roundoff error after multiplication as noise. Multiplying two B-bit numbers yields a $(2B - 1)$-bit number (assuming both numbers are signed). Further multiplications would keep extending the wordlength of the result unless a quantizer truncates or rounds it back down to B-bits after each multiply. The error that results from this "chopping" is modeled as an additive white noise source, uniformly distributed from $\frac{-2^{-B}}{2}$ to $\frac{2^{-B}}{2}$ (see Figure 4-1), assuming rounding is used to do the chopping. If truncation is used instead, then the error distribution will be from $[0, 2^{-B})$.

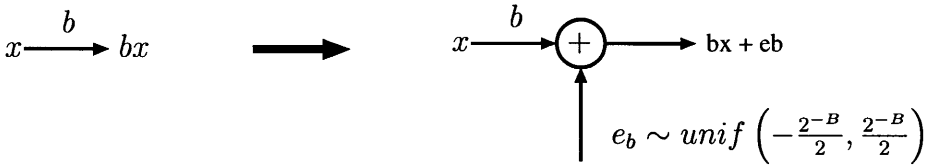

x
b
bx
x
b
$+$
bx + eb
$e_b \sim unif\left(-\frac{2^{-B}}{2}, \frac{2^{-B}}{2}\right)$

Figure 4-1: Multiplier with roundoff error modeled as noise.

Each noise source (one associated with each multiplier) is assumed to be uncorrelated with any others and with the input sequence quantization errors. Extensive testing over the years has borne out this model's validity (Roberts and Mullis [151, page 346] cite [20, 183] as references). [161] presented necessary and sufficient conditions for the *quantization* model to be valid. It is important to remember the distinction between *quantizer* noise and *roundoff* noise. Quantizer noise results from an A/D quantization while roundoff noise is the error in the B least significant bits after a mulitplication. Wong, in [189], gave sufficient conditions for the quantization error and roundoff noise to be uniformly distributed and white up to the first- and second-order moments, and for them to be mutually uncorrelated. He also investigated the effects of adding dithering to the input to the quantizer and concluded that adding dithering will almost always be adequate to guarantee the correctness of the white noise assumption. In [190], he specifically analyzed the roundoff noise in FIR filters. Barnes *et al.* [15] also studied the white noise assumption for the multiplicative error model.

In the many cases that do not satisfy these conditions, authors have presented persuasive arguments or experimental data to still justify the use of the uniform white noise model [185].

Essentially, the conditions require that the sequence is 'sufficiently exciting', i.e. it has a rich harmonic content and its amplitude spans several quantization steps. Also, the probability of overflow must be sufficiently low.

Stochastic noise models quantization error well not only in FWL cases, but also for A/D quantization [67, 69]. Stochastic noise could in fact potentially model any quantizer. Returning to our specific case, there are two methods to model roundoff error in the state-space model. Consider the state update equations

$$x_i[k+1] = \sum_{j=1}^{n} a_{ij}x_j[k] + b_iu[k]$$

We can quantize each subproduct $a_{ij}x_j[k]$ and $b_iu[k]$ and then sum them (rounding before summation), or we can quantize the entire sum (rounding after summation). This sometimes sloppy distinction in the literature deserves some extra space here.

Case 1:

Each subproduct is quantized. Thus,

$$x_i[k+1] = \sum_{j=1}^{n} Q[a_{ij}x_j[k]] + Q[b_iu[k]].$$

Applying the stochastic model for multipliers and assuming that all states have the same number of fractional bits, B_s, the variance of $x_i[k+1]$ will be $(n+1)\sigma^2$ where σ^2 is the variance of each noise source ($\frac{2^{-2B_s}}{12}$ in this case). This variance assumes that all a_{ij} and b_i cause roundoff error. In reality, 0, ±1, and positive integer coefficients with no fractional part do not cause any roundoff error. The output error variance, $\sigma^2_{\Delta y[k]}$, will also be $(n+1)\sigma^2$. Using a result from stochastic processes, the overall roundoff noise gain will be the variance of the input times the energy gain of the system,

$$\begin{aligned}\sigma_y^2 &= \left(n \sum_{i=1}^{n} \|g_i\|_2^2 + 1 \right) \sigma^2 \\ &= (n \operatorname{tr}(W_o) + n + 1)\sigma^2\end{aligned}$$

where, as before, g_i represent the impulse response sequence from the ith state variable to the output. The "+1" comes from quantization at the output node.

In the more general case, $0, \pm 1$ and integral coefficients do not induce roundoff error. Thus, a more accurate roundoff noise gain definition is

$$\sigma_y^2 = (\operatorname{tr}(QW_o) + I)\sigma^2$$

where Q is a diagonal matrix with $[Q]_{ii}$ equal to the number of non-integral coefficients on the ith row of $[A \;\; B]$.

Case 2:

The subproducts are not quantized until after they are summed.

$$x_i[k+1] = Q\left[\sum_{j=1}^{n} a_{ij} x_j[k] + b_i u[k] \right].$$

This model requires the use of a double precision accumulator. Since there is only one roundoff, the error variance of $x_i[k+1]$ is only σ^2. The overall roundoff noise gain will be

$$\begin{aligned}\sigma_y^2 &= \left(\sum_{i=1}^{n} \|g_i\|_2^2 + 1 \right) \sigma^2 \\ &= (\operatorname{tr}(W_o) + 1)\sigma^2\end{aligned}$$

This case is more appropriate for current high-end digital signal processors which provide double length (or longer) accumulators. Note also that in this case, a sparser realization with some $0, \pm 1$, and integral coefficients will *not* decrease the output variance as it would with Case 1 (though a sparse realization may offer other advantages such as increased speed/fewer operations). See Section 5.11 for Amit and Shaked's algorithm that generates sparse realizations.

Whichever case a particular implementation falls into, the literature defines the roundoff noise gain G as

$$G \triangleq \operatorname{tr}(W_o)$$

In solving many problems formulated with the roundoff noise model of Case 1, the simplifying assumption is made that *all* coefficients contribute to roundoff noise. In this case, minimizing roundoff noise gain for both cases reduces to minimizing G. Mullis and Roberts [122] and Hwang [73] first solved the minimum roundoff noise structure problem. See Section 5.6 for the optimal realization.

Note that for Case 1, minimizing $\operatorname{tr}(W_o)$ may not actually produce the *optimal* solution, i.e. the one that minimizes the sum $\operatorname{tr}(QW_o)+I$, which I shall call the *actual* roundoff noise. Increasing a diagonal element of W_o while significantly reducing one of Q may in fact lower the *actual* roundoff noise while increasing G.

Roundoff Noise Gain and Sensitivity

One would expect that a low sensitivity system would have low roundoff noise gain and vice versa. In fact, consider the two expressions:

$$\begin{aligned} \bar{M}_{L_{12}} &= \operatorname{tr}(W_c)\operatorname{tr}(W_o) + \operatorname{tr}(W_o) + \operatorname{tr}(W_c) \\ G &= \operatorname{tr}(W_o) \\ \Rightarrow \bar{M}_{L_{12}} &= G(\operatorname{tr}(W_c) + 1) + \operatorname{tr}(W_c) \end{aligned}$$

For an l_2-scaled system (with $\alpha = 1$), $\bar{M}_{L_{12}} = (n+1)G + n$. More generally, $\bar{M}_{L_{12}} = (\frac{n}{\alpha^2}+1)G + \frac{n}{\alpha^2}$. Since $G \geq 0$, those realizations that minimize G also minimize $\bar{M}_{L_{12}}$. Thus, the roundoff noise gain bounds sensitivity. Does sensitivity somehow bound roundoff noise gain also? The next section, which develops a stochastic sensitivity measure, provides an answer. I will also say more about this comparison in Section 5.6, where I discuss the realizations that minimize roundoff noise.

No one to my knowledge has developed the roundoff noise gain measure in the MIMO context. Note that the development of the synthetic measure in Section 4.7 shows the derivation of the noise gain of a MISO system.

4.6 Stochastic Models of Coefficient Quantization

Several authors, starting with Knowles and Olcayto [94, 132] and including Avenhaus [7], Crochiere [40], Moroney *et. al.* [119, 120], and more recently, Kawamata and Higuchi [91], Iwatsuki *et. al.* [77], and Gevers and Li [62] have proposed using a stochastic model for coefficient variation also. The basic idea is to assume that the coefficient quantization errors are uncorrelated random variables, uniformly distributed with zero mean and variance $\frac{2^{-2B_c}}{12}$. A justification for this is: At the earlier design stages, neither the coefficients nor the exact realization is known. In some sense, one can think of the 'ensemble' of structures under consideration. In fact, even the wordlength may not be known; in this case, though, it is not clear what the limits of the uniform distribution should be, and consequently what variance should be used. Kawamata and Higuchi present experimental evidence that also supports this stochastic model.

Knowles and Olcayto [94] first proposed this idea with the following measure

$$\sigma^2 = E\left[\frac{1}{2\pi}\int_0^{2\pi} \left|H(e^{j\omega}) - H^*(e^{j\omega})\right|^2 \, d\omega\right] = E\left[\frac{1}{2\pi}\int_0^{2\pi} \left|\Delta H(e^{j\omega})\right|^2 \, d\omega\right]$$

where $\Delta H(z)$ models the errors with the following first-order approximation

$$\Delta H(z) = \sum_{i=1}^{n} \frac{\partial H(z)}{\partial a_i} (\Delta a_i)$$

Avenhaus [7] suggests using the variance of the error in the transfer function *magnitude* ($|H(e^{j\omega})|^2$) as a measure. If $H_{mag} = |H(e^{j\omega})|^2$ and $H^*_{mag} = |H^*(e^{j\omega})|^2$, then with a first order approximation,

$$\begin{aligned} H^*_{mag} &= H_{mag} + \Delta H_{mag} \\ &= H_{mag} + \sum_i \frac{\partial H_{mag}}{\partial a_i} \Delta a_i \end{aligned}$$

Crochiere [40] furthers Avenhaus' development and uses this stochastic model specifically to design filters to match frequency response magnitude specifications. He presents synthesis methods that result in filter designs meeting magnitude specifications with a minimal wordlength. Given a filter design, $T(z)$, he uses a "statistical wordlength" to compute the minimum wordlength required for the filter to meet the magnitude specifications with some tolerance. This tolerance term was mentioned earlier in the context of frequency weighted sensitivity functions. Crochiere's work easily extends to functions other than just magnitude.

Moroney *et al.* [119,120] applied the stochastic model in the LQG context. In the process, they derive expressions for the second-order sensitivity, with respect to the realization coefficients, of a function of the coefficients (for example, LQG cost).

Kawamata and Higuchi [91] were the first to apply the stochastic quantization error model in the state-space. They were also the first to analyze *output* error due to coefficient quantization, as most earlier work considered coefficient quantization in the context of sensitivity while measuring output error performance with roundoff noise. They develop expressions using both a deterministic and stochastic model. The stochastic model, developed with a first-order approximation, results in an output variance expression that is equivalent to the roundoff noise gain (of Case 1 in Section 4.5). Kawamata and Higuchi define statistical sensitivity as $S_c \triangleq \operatorname{tr}(QW_o)(= G)$. Thus, they complete the circle and show that minimizing sensitivity is equivalent to minimizing roundoff noise if one uses a first-order approximation and a stochastic quantization error model. That roundoff noise bounds sensitivity was known before (see Section 4.1), since $\bar{M}_{L_{12}}$ is a multiple of G. Kawamata and Higuchi's result [91] now relates coefficient quantization error (as measured by a stochastic sensitivity function) to roundoff noise gain.

In [77], they extend the work in [91] to take into account coefficients that do not have any quantization error, i.e. $x = Q[x]$.

In the controls context, Fialho and Georgiou [54] develop a stability robustness measure, also assuming stochastic coefficient quantization (see Section 4.10).

Gevers and Li also use a stochastic coefficient quantization error model in their synthetic measures (one for filters and one for controllers), which unifies the measurement of the effects of coefficient quantization and roundoff noise into one performance metric. I elaborate on this in the next section.

4.7 A Synthetic Measure

Gevers and Li [62, Chapter 7] develop a synthetic measure that combines state quantization noise and coefficient sensitivity. The unified treatment (albeit stochastic) of roundoff noise and coefficient quantization results in a weighted measure, the synthetic noise gain. The interesting development of this measure ties together the M_{L_2} metric, the stochastic coefficient quantization measures, and roundoff noise.

Starting with the model description that takes into account all FWL effects (equation (2.8), listed here again for convenience)

$$\begin{aligned} x_Q^*[k+1] &= A^* Q[x_Q^*[k]] + B^* Q[u[k]] \\ y_Q^*[k] &= C^* Q[x_Q^*[k]] + D^* Q[u[k]] \end{aligned} \tag{4.31}$$

the roundoff noise is

$$e_x[k] \triangleq x_Q^*[k] - Q[x_Q^*[k]]$$

Including the roundoff noise using the uniform, white noise model in (4.31) and assuming that the input $u[k]$ has zero quantization error,

$$\begin{aligned} x_Q^*[k+1] &= A^* x_Q^*[k] + B^* u[k] - A^* e_x[k] \\ y_Q^*[k] &= C^* x_Q^*[k] + D^* u[k] - C^* e_x[k] \end{aligned}$$

Define the degradation in output as the difference

$$\begin{aligned} \Delta y[k] &= y[k] - y_Q^*[k] = (y[k] - y^*[k]) + (y^*[k] - y_Q^*[k]) \\ &= \Delta y^*[k] + \Delta y_Q^*[k] \end{aligned}$$

The first term can be thought of as the error due to coefficient quantization and the second term, the roundoff error due to state quantization. The input $u[k]$ drives the output $\Delta y^*[k]$, while the roundoff noise $e_x[k]$ drives the output $\Delta y_Q^*[k]$. Note that the white-noise roundoff error model allows this neat separation of the different errors. Since these two inputs are assumed to be independent and since both outputs are zero-mean, the output error variance can be separated as

$$\sigma^2_{\Delta y[k]} = \sigma_1^2 + \sigma_2^2$$

where σ_1^2 is the variance due to coefficient quantization and σ_2^2 is the variance due to roundoff noise.

The error due to the first term is

$$\begin{aligned}\Delta y^*[k] &= y[k] - y^*[k] = (h[k] - h^*[k]){*}u[k]\\ &= \Delta h[k]{*}u[k]\end{aligned}$$

which, in the frequency domain and with a first order approximation,

$$\Delta H(z) = H(z) - H^*(z) = \sum_{i=1}^{n} \frac{\partial H(z)}{\partial c_i}\,(\Delta c_i)$$

where the c_i denote the coefficients in $H(z)$. Adopting the stochastic coefficient quantization model developed in Section 4.6, each of the Δc_i will be a uniformly distributed random variable with zero mean and variance $\frac{2^{-2B_c}}{12}$. Also, they are all uncorrelated with each other. Then,

$$E[(\Delta y^*[k])^2] = \sum_{i=1}^{n} \left(\frac{1}{2\pi} \int_0^{2\pi} \left(\frac{\partial H(e^{-j\omega})}{\partial c_i} \right)^T \frac{\partial H(e^{j\omega})}{\partial c_i}\, d\omega \right) \sigma_c^2$$

where σ_c^2 is the variance of the Δc_i. All the cross terms drop since the Δc_i are zero-mean and uncorrelated.

In the state-space setting, the c_i are entries of the system matrices, so the above simplifies to

$$\begin{aligned}E[(\Delta y^*[k])^2] &= \left[\frac{1}{2\pi} \int_0^{2\pi} \left\| \frac{\partial H(e^{j\omega})}{\partial A} \right\|_F^2 + \left\| \frac{\partial H(e^{j\omega})}{\partial B} \right\|_F^2 + \left\| \frac{\partial H(e^{j\omega})}{\partial C} \right\|_F^2 d\omega + \left\| \frac{\partial H(z)}{\partial D} \right\|_F^2 \right] \sigma_c^2\\ &= \left\| \frac{\partial H(z)}{\partial A} \right\|_2^2 + \left\| \frac{\partial H(z)}{\partial B} \right\|_2^2 + \left\| \frac{\partial H(z)}{\partial C} \right\|_2^2 + \left\| \frac{\partial H(z)}{\partial D} \right\|_2^2\\ &= \operatorname{tr}(W_A + W_c + W_o) + 1\\ &= (M_{L_2} + 1)\sigma_c^2 \end{aligned} \qquad (4.32)$$

As mentioned before, since the goal is to ultimately search for the optimal transformation to minimize this measure, the coordinate independent D term is not useful. We can therefore reduce (4.32) to

$$\sigma_1^2 = M_{L_2}\sigma_c^2 \tag{4.34}$$

This reappearance of the M_{L_2} measure with a stochastic coefficient quantization model confirms that it is a more appropriate measure of sensitivity than $M_{L_{12}}$.

Now, to compute the variance due to the second term, we must first compute $y^*[k] - y_Q^*[k]$:

$$\begin{aligned} y'[k] = y^*[k] - y_Q^*[k] &= C^*(x^*[k] - Q[x_Q^*[k]]) \\ &= C^*(x^*[k] - (x_Q^*[k] - e_x[k])) \\ &= C^*(x^*[k] - x_Q^*[k]) + C^*e_x[k] \end{aligned}$$

Define a new state vector, $x'[k] = x^*[k] - x_Q^*[k]$. Then

$$\begin{aligned} x'[k+1] &= A^*x'[k] + A^*e_x[k] \\ y'[k] &= C^*x'[k] + C^*e_x[k] \end{aligned}$$

This system is MISO. Thus, we will have to carry out the noise analysis again (since the earlier roundoff noise gain calculations assumed a SISO system). Note that

$$x'[k] = \sum_{i=0}^{\infty}(A^*)^i A^* e_x[k-1-i]$$

Substituting into the output equation,

$$\begin{aligned} y'[k] &= C^*x'[k] + C^*e_x[k] \\ &= C^*\left[\sum_{i=0}^{\infty}(A^*)^i A^* e_x[k-1-i]\right] + C^*e_x[k] \end{aligned}$$

Computing the steady-state error variance,

$$\sigma_2^2 = \left\{\mathrm{tr}\left[(A^*)^T W_o^* A^*\right] + C^*(C^*)^T\right\}\sigma_s^2$$

The noise gain of this system will be $\mathrm{tr}\,(W_o^*)\sigma_s^2$, where W_o^* is the observability Gramian for the pair (A^*, C^*) and σ_s^2 is the variance for each signal roundoff and is equal to $\frac{2^{-2B_s}}{12}$.

Combining σ_1^2 and σ_2^2,

$$\begin{aligned}\sigma^2 &= \sigma_1^2 + \sigma_2^2 \\ &= M_{L_2}\sigma_c^2 + \mathrm{tr}\,(W_o^*)\sigma_s^2\end{aligned}$$

Dividing by σ_c^2 and approximating $\mathrm{tr}\,(W_o^*)$ by $\mathrm{tr}\,(W_o)$, we get the synthetic *total noise gain*, G_T.

$$\begin{aligned}G_T &\triangleq \mathrm{tr}\,(W_A + W_o + W_c) + \frac{\sigma_s^2}{\sigma_c^2}\mathrm{tr}\,(W_o) \\ &= M_{L_2} + \rho^2 G \qquad (4.35)\end{aligned}$$

where $\rho^2 = \frac{\sigma_s^2}{\sigma_c^2} = 2^{2(B_c - B_s)}$. This weighted measure ties together the two FWL effects on linear systems into one. The weighting factor ρ allows the designer to emphasize one type of error over the other.

A larger B_c will cause ρ to get larger and favor roundoff noise minimization while the converse will favor minimizing coefficient quantization error. Note that if $\rho = 0$, we recover the measure for coefficient quantization, and if $\rho = \infty$, we recover the roundoff noise gain measure (in the sense that minimizing G_T in this case will yield the same solution as minimizing the roundoff noise gain).

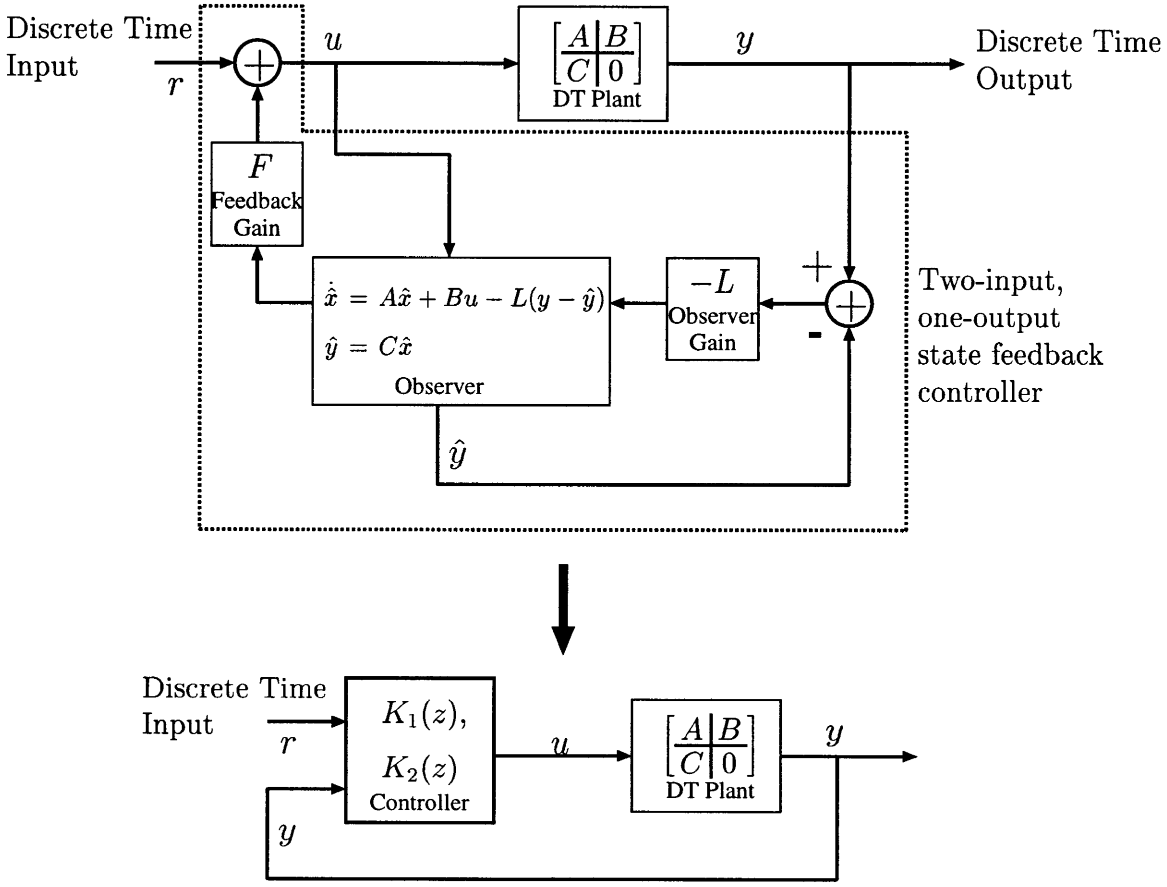
The realization that minimizes this measure is given in Section 5.7.

4.8 The Controls Context

As mentioned earlier, in the FWL controls problem, the goal is (or at least should be) to minimize the sensitivity of the *entire closed loop* to coefficient perturbations and roundoff errors in the controller. As such, the above measures are not adequate and need to be extended. Ideally, since most controllers are used with CT plants, one would like to include the CT model in the minimization problem. However, the mixing of the CT/DT domains leads to more difficulties, which have only recently been addressed. Prior to that, most FWL controls problems assumed a discretized plant (discretized at the sampling frequency of the controller) and presented a solution for this problem.

Following the layout earlier in the chapter, I present the deterministic sensitivity measures first and then the stochastic ones.

I will briefly set up the controls context along with some of the machinery to address it.



$$u(t) = K_1(z)r(t) - K_2(z)y(t)$$

Figure 4-2: Feedback-based state estimate observer.

The plant model is

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] \end{aligned} \tag{4.36}$$

The observer to estimate the states of the plant evolves according to

$$\begin{aligned}\hat{x}[k+1] &= A\hat{x}[k] + Bu[k] - L(y - \hat{y})\\ \hat{y}[k] &= C\hat{x}[k]\end{aligned} \tag{4.37}$$

with an observer gain L. Combining a static feedback law of the form $u[k] = F\hat{x}[k] + r[k]$ (F is the feedback gain matrix) with (4.36) and (4.37) results in

$$\begin{aligned}x[k+1] &= Ax[k] + B(F\hat{x}[k] + r[k])\\ y[k] &= Cx[k]\end{aligned} \tag{4.38}$$

and

$$\begin{aligned}\hat{x}[k+1] &= A\hat{x}[k] + B(F\hat{x}[k] + r[k]) - L(Cx - C\hat{x})\\ \hat{y}[k] &= C\hat{x}[k]\end{aligned} \tag{4.39}$$

I will assume that both the F and L matrices were designed using standard methods like pole-placement and/or solving the Kalman design equations.

Combining the two state vectors,

$$\begin{aligned}\begin{bmatrix} x[k+1] \\ \hat{x}[k+1] \end{bmatrix} &= \begin{bmatrix} A & BF \\ -LC & A+BF+LC \end{bmatrix} \begin{bmatrix} x[k] \\ \hat{x}[k] \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r[k] \qquad (4.40)\\ \begin{bmatrix} y[k] \\ \hat{y}[k] \end{bmatrix} &= \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x[k] \\ \hat{x}[k] \end{bmatrix}\end{aligned}$$

Changing coordinates to the standard observer form where $\tilde{x} = x - \hat{x}$ is the state error vector,

$$\begin{aligned}\begin{bmatrix} x[k+1] \\ \tilde{x}[k+1] \end{bmatrix} &= \begin{bmatrix} A+BF & -BF \\ 0 & A+LC \end{bmatrix} \begin{bmatrix} x[k] \\ \tilde{x}[k] \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r[k]\\ \begin{bmatrix} y[k] \\ \tilde{y}[k] \end{bmatrix} &= \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x[k] \\ \tilde{x}[k] \end{bmatrix}\end{aligned} \tag{4.41}$$

Equation (4.41) displays the celebrated 'separation principle'. Since the eigenvalues of $(A + BF)$ and $(A + LC)$ sepearately comprise the eigenvalues of the closed-loop system, one can independently design the corresponding feedback and observer gain matrices.

That is the development in the infinite precision case. However, introducing coefficient quantization breaks down the 'separation principle'. Let F_o, L_o denote the quantized versions of F and L while A_o, B_o, C_o denote the finite precision system matrices of the observer. Substituting into (4.38) and (4.39), the error equation becomes

$$\begin{aligned}\dot{\tilde{x}} &= \dot{x} - \dot{\hat{x}} \\ &= Ax + BF_ox + Br - A_o\hat{x} - B_oF_o\hat{x} - B_or + L_o(Cx - C_o\hat{x}) \\ &= (A - L_oC)x - (A_o - L_oC_o)\hat{x} + \Delta BF_o\hat{x} + \Delta Br\end{aligned}$$

and the overall state space transforms to

$$\begin{bmatrix} x[k+1] \\ \tilde{x}[k+1] \end{bmatrix} = \begin{bmatrix} A + BF_o & -BF_o \\ \Delta A + L_o\Delta C + \Delta BF_o & A_o + L_oC_o + \Delta B_oF_o \end{bmatrix} \begin{bmatrix} x[k] \\ \tilde{x}[k] \end{bmatrix} + \begin{bmatrix} B \\ B_o \end{bmatrix} r[k]$$

$$\begin{bmatrix} y[k] \\ \tilde{y}[k] \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & C_o \end{bmatrix} \begin{bmatrix} x[k] \\ \tilde{x}[k] \end{bmatrix}$$

The new cross term $\Delta A + L_o\Delta C + \Delta BF_o$ couples the plant and the error dynamics of the observer. Thus, the 'separation principle' breaks down, and pole-placement for the plant and for the error dynamics *cannot* be done independently. Note that if $\Delta A, \Delta B, \Delta C$ are all zero, we recover the original plant and error dynamics.

The immediate question is how to measure (and then minimize) the effects of coefficient quantization. The second question, of how to measure (and minimize) roundoff noise due to state quantization (which is completely ignored in the above equations), will continue to lurk in the background for the moment. Before proceeding, it is important to reiterate the point made earlier: we can minimize the sensitivity of F, L to coefficient quantization by finding a different *realization* using similarity transformations, or we can *redesign* F, L taking into account FWL effects and the *entire* closed loop.

The first approach seeks to find the infinite-precision coefficients of F and L using the standard design equations (pole-placement etc.) and then to find a transformation to minimize the sensitivity of the resulting realization over the set of all equivalent realizations. The latter would probably require a stochastic approach to coefficient quantization. Currently, there is no known deterministic or stochastic solution for doing feedback/observer *design* while including coefficient quantization errors.

I will present a coefficient perturbation sensitivity measure that Gevers and Li included in [62]. Their solution is of the former type – it returns the optimal similarity transformation for the given F, L. They also mention that one could set up and probably solve a (hard) H_∞ problem to find a solution to the latter design problem.[6]

Much of the material in the following section draws heavily from Gevers and Li's text [62]. They are in the minority to consider FWL effects in pole-placement outside the LQG framework.

4.9 Closed-Loop Sensitivity

The quantity $M_{cl,L_{12}}$ measures closed loop sensitivity to perturbations in the compensator coefficients (with a discretized version of the CT plant) [62]. Based on the picture and machinery developed above, we shall proceed with writing the $M_{cl,L_{12}}$ sensitivity. Using (4.40), let

$$\bar{A} = \begin{bmatrix} A & BF \\ -LC & \Phi \end{bmatrix}; \qquad \bar{B} = \begin{bmatrix} B \\ B \end{bmatrix}; \qquad \bar{C} = [C \; 0]$$

where the output of $(\bar{A}, \bar{B}, \bar{C})$ is $y[k]$ and $\Phi = A + BF + LC$. The closed-loop transfer function $H_{cl}(z)$ is then

$$H_{cl}(z) = \bar{C}(zI - \bar{A})^{-1}\bar{B}$$

The resulting sensitivity measure then is

$$M_{cl,L_{12}} = \left\| \frac{\partial H_{cl}(z)}{\partial \Phi} \right\|_1^2 + \left\| \frac{\partial H_{cl}(z)}{\partial B} \right\|_2^2 + \left\| \frac{\partial H_{cl}(z)}{\partial L} \right\|_2^2 + \left\| \frac{\partial H_{cl}(z)}{\partial F} \right\|_2^2$$

The reader is referred to [62, chapter 9] for details on the computations of the measure.

[6] Such a problem description would be: Find the controller to minimize the H_∞ norm of the difference in the transfer functions in the setup of Figure 3-7

Closed-Loop Sensitivity Using a Continuous Model of the Plant

Madievski *et al.* [111] developed an operator-based measure which allows a time-domain analysis. It does not require discretization of the CT plant at the sampling frequency of the controller. When the measure is ultimately evaluated, a discrete approximation of the CT plant is used, but the approximation's behavior can be made arbitrarily close to that of the CT plant (i.e. the discretization rate is not restricted to be the sampling rate of the controller). Such a discretization allows one to practically eliminate intersample effects.

The system setup is as shown in Figure B-2, repeated here in Figure 4-3 for convenience.

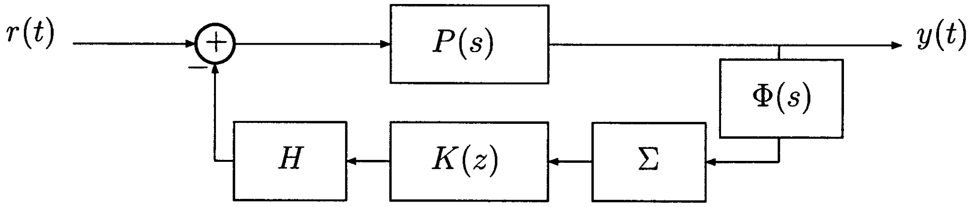


Figure 4-3: A hybrid system.

(In the MIMO case, Φ, H, Σ will be diagonal operators.) The closed-loop linear periodically time-varying operator $\mathcal{H}$ from $r(t)$ to $y(t)$ is

$$\mathcal{H} = PHK\Sigma\Phi(I + PHK\Sigma\Phi)^{-1} \tag{4.42}$$

with an associated causal impulse response $\mathcal{H}(t, s)$ such that

$$\begin{aligned} y(t) &= \int_{-\infty}^{t} \mathcal{H}(t,s)u(s)\ ds \\ \mathcal{H}(t+\tau, s+\tau) &= \mathcal{H}(t,s) \end{aligned}$$

The derivative of (4.42) with respect to a coefficient a in the controller K is

$$\frac{\partial \mathcal{H}}{\partial a} = \mathcal{V}\frac{\partial K}{\partial a}\mathcal{W}$$

where

$$\mathcal{V} = (I + PHK\Sigma\Phi)^{-1}PH$$

$$\mathcal{W} = \Sigma\Phi(I + PHK\Sigma\Phi)^{-1}$$

Note that $\mathcal{V}$ and $\mathcal{W}$ can be thought of simply as operators between different input/output points in the closed-loop.

Then,

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial a_{i,j}} &= (I_L + PHK\Sigma\Phi)^{-1}PHC(zI_R - A)^{-1}e_ie_j^T(zI_R - A)^{-1}B\Sigma\Phi(I_L + PHK\Sigma\Phi)^{-1} \\ &= \mathcal{V}_A e_i e_j^T \mathcal{W}_A\end{aligned}$$

where

$$\begin{aligned}\mathcal{V}_A &= \mathcal{V}C(zI_R - A)^{-1} \\ \mathcal{W}_A &= (zI_R - A)^{-1}B\mathcal{W}\end{aligned}$$

$\mathcal{V}_A$ and $\mathcal{W}_A$ are also stable operators between different input/output points in the closed-loop. Similarly,

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial b_{i,j}} &= \mathcal{V}_A e_i e_j^T \mathcal{W} \\ \frac{\partial \mathcal{H}}{\partial c_{i,j}} &= \mathcal{V} e_i e_j^T \mathcal{W}_A\end{aligned}$$

Applying Graham's First Transformation Principle [143],

$$\begin{aligned}\frac{\partial \mathcal{H}_{k,l}}{\partial A} &= \mathcal{V}_A^T e_k e_l^T \mathcal{W}_A^T \\ \frac{\partial \mathcal{H}_{k,l}}{\partial B} &= \mathcal{V}_A^T e_k e_l^T \mathcal{W}^T \\ \frac{\partial \mathcal{H}_{k,l}}{\partial C} &= \mathcal{V}^T e_k e_l^T \mathcal{W}_A^T\end{aligned}$$

Finally, using these operator derivatives, we can define the measure M_2:

$$M_2 = \sum_{k,l}\left(\left\|\frac{\partial \mathcal{H}_{k,l}}{\partial A}\right\|_2^2 + \left\|\frac{\partial \mathcal{H}_{k,l}}{\partial B}\right\|_2^2 + \left\|\frac{\partial \mathcal{H}_{k,l}}{\partial C}\right\|_2^2\right)$$

The norms are not induced norms, but rather "simply associated with the impulse response representation of a stable operator" and defined as

$$\|\mathcal{U}\|_2^2 = \left[\int_0^T dt \int_{-\infty}^t \|\mathcal{U}(t,s)\|_F^2 \, ds\right]$$

where $\mathcal{U}(t,s)$ is a periodic matrix impulse response defined in the half-plane $s \leq t$ and $\mathcal{U}(t+\tau, s+\tau) = \mathcal{U}(t,s)$. Because of the periodicity of $\mathcal{U}(t,s)$, this norm takes into account all values of $\mathcal{U}$ without requiring integration with respect to t from $(-\infty, \infty)$.

$\mathcal{U}$ should have the (exponential) stability property

$$||\mathcal{U}(t,s)||_F \leq \alpha e^{-\beta(t-s)}$$

for some $\alpha, \beta > 0$.

Madievski *et. al.* [111] importantly note that this measure is "intrinsically a time-domain rather than a frequency-domain one." Moreover, just as Gevers and Li [62] did, they use an $\mathcal{L}_2$ norm for the first term, which makes much more sense due to the relation between the time and frequency domain of the norms, via Parseval's relation.

They do not however impose a scaling constraint on the controller coefficients. It is not obvious how one would impose a scaling constraint with an operator representation.

Evaluating the norms above is difficult, but an algorithm is given in [111]. The key idea is the approximation of the CT plant with a fast-sampled discretization (based on Keller and Anderson's work [92]). See Section 5.9.3 for more details on fast discretization and [111] for proofs and further details with regard to the M_2 measure.

The work to evaluate and minimize the M_{L_2} measure for MIMO filters is also embedded within [111]. Note that the MIMO filter sensitivity measure will be a simpler version of the M_2 measure given.

4.10 Stability Robustness Measures

Recently, Li [100] and Fialho and Georgiou [54, 55] have treated the FWL problem as a stability robustness problem. The "stability robustness" problem is the same one mentioned before: How much (as measured with an appropriate norm) can we perturb the controller parameters and still maintain closed-loop stability? Note, however, that before we were concerned with filter stability, not closed-loop stability. The term stability robustness is used more in the modern controls paradigm, with all the machinery of robust controls.

I will briefly set the problem up here and then describe Li's deterministic approach. Fialho and Giorgiou's stochastic approximation appears after that.

Consider a general DT closed-loop system, with a discretized CT component (usually composed of a CT plant and an anti-aliasing filter) $(A_c, B_c, C_c, 0)$ and a DT controller $K = (A_d, B_d, C_d, D_d)$. The closed-loop system's 'A'-matrix as a function of the controller K will be

$$\begin{aligned} A(K) &= \begin{bmatrix} A_c & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} B_c & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} D_d & C_d \\ B_d & A_d \end{bmatrix} \begin{bmatrix} C_c & 0 \\ 0 & I \end{bmatrix} \\ &\triangleq M_0 + M_1 K M_2 \end{aligned}$$

The controller parameters are all in the K matrix. Perturbing K to $K + \Delta$ will change $A(K)$ to $A(K) + M_1 \Delta M_2$. The stability robustness problem asks for what set of perturbations Δ one can guarantee stability. To mathematically formulate the problem, we need to first decide how to measure the magnitude or 'energy' of a perturbation matrix Δ. This magnitude will be the appropriate norm that defines the "how much" in this section's opening question. One measure of magnitude, which I will call $\mu(\cdot)$, is simply the maximum of all the matrix entries:

$$\mu(\Delta) \triangleq \max_{i,j} |\Delta_{ij}|$$

Then, the stability robustness problem is to find the largest perturbation, call it $\mu_0(K)$, that will not destabilize the closed-loop. Equivalently, $\mu_0(K)$ is the smallest perturbation that will destabilize the closed-loop.

$$\mu_0(K) \triangleq \inf\{\mu(\Delta) | A(K) + M_1 \Delta M_2 \text{ is unstable.}\}$$

The resulting μ_0 gives the following guarantee: Given a specific controller K, no matter which Δ is applied, as long as all of its entries are less than the bound $\mu_0(K)$, one can guarantee that $A(K) + M_1 \Delta M_2$ is stable. The general solution to this minimization problem with this norm is unknown.

A slightly simpler formulation with a different norm does have a known solution. The *real stability radius* problem is defined as:

$$r_R(K) \triangleq \inf\{\|\Delta\|_2 \, | A(K) + M_1 \Delta M_2 \text{ is unstable.}\}$$

where $||\Delta||_2$ is the induced matrix 2-norm (equivalently, the maximum singular value, $\sigma_{max}(\Delta)$). What this bound says is that, given a specific controller K, no matter which Δ is applied, one can guarantee that $A(K) + M_1 \Delta M_2$ is stable as long as its maximum singular value is less than $r_R(K)$. Note that since $\max_{i,j} |\Delta_{ij}| \leq ||\Delta||_2$, $\mu_0(K) \leq r_R(K)$.[7] In general, the two bounds can be quite far apart, depending on the exact size and structure of Δ [55, 100].

Restricting the perturbations to be FWL quantizations gives much more useful and pertinent information. If $\mu_0(K) > \frac{2^{-B_c}}{2}$, then μ_0's exact value does not matter. Stability is always guaranteed since all entries of Δ are always less than $\frac{2^{-B_c}}{2}$. By definition of $r_R(K)$, if $||\Delta||_2$, which is a function of the wordlength B_c, is less than $r_R(K)$, then stability is also guaranteed.

Li continues in a more interesting direction. He poses the problem: What similarity transformation, T, applied to K will maximize performance (which can be measured in different ways)? If performance is measured in terms of the stability robustness bound, then the optimization problem translates to finding the transformation that will allow the entries of Δ to be large but still will not destabilize the closed-loop system. Mathematically,

$$\sup_{\{T | K = (T^{-1}AT, T^{-1}B, CT, D)\}} \{\mu_0(K)\}$$

The real stability radius problem translates to finding the T such that

$$\sup_{\{T | K = (T^{-1}AT, T^{-1}B, CT, D)\}} \{r_R(K)\}$$

Both $\mu_0(K)$ and $r_R(K)$ change in a very complicated way as K varies over the set of equivalent realizations, and the solution of neither of the above problems is known. Instead of directly trying to solve either problem, Li uses eigenvalue sensitivity along with these robustness measures to develop a tractable function of K which he then minimizes. The presentation given here follows [100].

[7] The proof is as follows: Let $a_{kl} = \max_{i,j} |A_{ij}|$. $\sigma_{max}(A) = \max_{||x||_2 = 1} ||Ax||_2$. Then, choose $x = e_l \Rightarrow \sigma_{max}(A) \geq ||Ax||_2 = \left(\sum_{i=1}^{n} a_{il}^2 \right)^{1/2} \geq a_{kl}$

The *stability margin*, m, of a stable eigenvalue λ of A is the smallest distance between λ and the unit circle: $m \triangleq 1 - |\lambda|$, i.e. the pole must be perturbed by at least magnitude m for it to become an unstable pole. Choose a perturbation Δ such that $\mu(\Delta K) = \mu_0(K)$. Then, by definition of μ_0, one of the poles (call it λ_k) must be unstable and has been moved by more than m_k. Thus, for this pole, $m_k \leq |\lambda_k(A + \Delta A) - \lambda_k(A)|$.

Li makes the following assumption: $\mu_0(K)$ is small. If $\mu_0(K)$ were large, the controller structure would not matter much (at least as far as measuring stability robustness) since all structures would be stable. Thus, it makes sense to assume $\mu_0(K)$ is small. Using a first-order approximation for the eigenvalue perturbation,

$$\begin{aligned} m_k &\leq \left| \sum_{i,j} \frac{\partial \lambda_k}{\partial c_{ij}} \Delta c_{ij} \right| \\ &= \sqrt{\left(\sum_{i,j} \frac{\partial \lambda_k}{\partial c_{ij}} \Delta c_{ij} \right)^2} \\ &\leq \sqrt{\mu_0(K)^2 (\sum_{i,j} \frac{\partial \lambda_k}{\partial c_{ij}})^2} \\ &= \mu_0(K) \sqrt{(\sum_{i,j} \frac{\partial \lambda_k}{\partial c_{ij}})^2} \\ &\leq \mu_0(K) \sqrt{N \sum_{i,j} (\frac{\partial \lambda_k}{\partial c_{ij}})^2} \end{aligned}$$

where c_{ij} are the coefficients of the controller K. The last inequality is due to the fact that $\left(\sum_{i=1}^{n} a_i \right)^2 \leq n \sum_{i=1}^{n} a_i^2$. Tightening the bound by accounting for $0, \pm 1$ coefficients,

$$m_k \leq \mu_0(K) \sqrt{N \sum_{i,j} \delta(c_{ij}) (\frac{\partial \lambda_k}{\partial c_{ij}})^2}$$

where $\delta(c_{ij}) = 0$ for $c_{ij} = 0, \pm 1$, else $\delta(c_{ij}) = 1$, and N is the number of non-zero elements in K. Note that one could also take into account integer coefficients by setting $\delta(c_{ij})$ to 0 for those cases too. The remainder of the presentation does *not* factor in this extension.

Summing over all the poles,

$$\mu_0(K) \geq \sqrt{\frac{m_k^2}{N\Psi_{pk}}} \triangleq \zeta_k$$

where Ψ_{pk}, the pole sensitivity (4.22), is

$$\Psi_{pk} = \sum_{i,j=1} \delta(c_{ij}) \left(\frac{\partial \lambda_k}{\partial c_{ij}}\right)^2$$

Li then defines the new measure $\mu_0^*(K)$ as

$$\mu_0^*(K) \triangleq \min_k \zeta_k$$

From the definition of $\mu_0(K)$, it immediately follows that the system is stable if

$$\mu(\Delta K) \leq \mu_0^*(K)$$

Using the equations of Section 4.3.1, one can compute pole sensitivity and hence the measure $\mu_0^*(K)$. The only difference between the computations in Section 4.3.1 and the ones in this section is that the ones here are for a controller, not for a filter as before. The partials that are now necessary are $\dfrac{\partial \lambda_k}{\partial A_d}, \dfrac{\partial \lambda_k}{\partial B_d}, \dfrac{\partial \lambda_k}{\partial C_d}$, and $\dfrac{\partial \lambda_k}{\partial D_d}$.

Li then develops the change in the measure under different transformations with the goal of maximizing it. He restricts the search space to the space of similarity transformations for a given controller, K_0, i.e.

$$\max_{\{T|detT\neq 0\}} \mu_0^*(K_0)$$

See [100] for the optimal transformation.

Another contribution of [100] is an extension and adaption of Amit and Shaked's $0, \pm 1$ algorithm for sparse controller design (see Section 5.11 for a discussion of this algorithm).

Refer to [100] for details and proofs.

The Stochastic Coefficient Quantization Approach

Fialho and Georgiou [55] investigate the minimum wordlength required to maintain stability. They use the following theorem: Assuming no unstable pole/zero cancellation between the plant and the anti-aliasing filter and assuming a non-pathological sampling rate [36], the continuous-time feedback system $[P, HK\Sigma\Phi]$ is L_2 input-output stable if and only if the discrete-time feedback system $[\Sigma\Phi PH, K]$ is stable. Thus, a discretization of the CT component, ΦP, does not in any way change the stability, i.e. some method that included the CT components as CT components instead of discrete-time approximations would get no different or better stability results.

Fialho and Georgiou use the mean and variance of $||\Delta||_F$ to bound the wordlength:

$$E[||\Delta||_F^2] = N\frac{2^{-2B_c}}{12}; \qquad \sigma^2_{||\Delta||_F^2} = N\frac{2^{-4B_c}}{12}$$

where N is the number of non-zero entries in Δ. Any integer entry will not contribute to quantization error[8]. Applying the Central Limit Theorem, $||\Delta||_F^2$ will be normally distributed with the above mean and variance. This gives a distribution of an upper bound of $||\Delta||_2$ as a function of the quantization step. Then, the minimum wordlength $B_{c,min}$ is

$$B_{c,min} = \left\lceil \log_2 \frac{2\sqrt{\frac{N}{12} + \sqrt{\frac{N}{45}}}}{r_R(X)} \right\rceil$$

They also solve for the minimal wordlength needed for performance robustness. Refer to [55] for details.

4.11 Closed-Loop Noise Gain

Just as the $M_{cl,L_{12}}$ measure parallels $M_{L_{12}}$, the closed-loop roundoff noise gain, G_{cl}, developed in this section will parallel roundoff noise gain, G, developed earlier for digital filters.

First, I will develop the closed-loop equations for the system with roundoff noise. I will assume that the system matrices are realized without any coefficient quantization error. Define $\hat{x}^*$ to be the rounded observer state and x^* and y^* to be the infinite-precision state and output of the CT system evolving with $Q[u[k]]$ instead of $u[k]$

[8] Actually, any coefficient that can be realized exactly will not contribute.

and $Q[r[k]]$ instead of $r[k]$. Let $u[k] = F\hat{x}[k] + r[k]$ as before. To ease the notational strain, for this example I will use $\dot{\hat{x}} \equiv \hat{x}[k+1]$ and $\hat{x} \equiv \hat{x}[k]$ for all variables.

First, define $e_x \triangleq \hat{x}^* - Q[\hat{x}^*]$. Then,

$$\begin{align*}
u &= F(\hat{x} + e_x) + (r + e_r) \\
\dot{\hat{x}}^* &= A(\hat{x}^* - e_x) + Bu - L((y^* - e_y) - (\hat{y}^* - e_{\hat{y}}) \\
&= A(\hat{x}^* - e_x) + B(F(\hat{x}^* - e_x) + (r - e_r)) - L((y^* - e_y) - (\hat{y}^* - e_{\hat{y}}) \\
&= (A + BF)\hat{x}^* - (A + BF)e_x + Br - Be_r - L(Cx^* - e_y) + L(C(\hat{x}^* - e_x) \\
&= (A + BF + LC)\hat{x}^* - (A + BF + LC)e_x + Br - Be_r - L(Cx^* - e_y) \\
&= \Phi\hat{x}^* + Br - [\,\Phi \;\, B \;\, -L \;\, 0\,] \begin{bmatrix} e_x \\ e_r \\ e_y \\ e_u \end{bmatrix} \tag{4.43}
\end{align*}$$

and

$$\begin{align*}
\dot{x}^* &= Ax^* + BQ[u] \\
&= Ax^* + B(F(\hat{x}^* - e_x) + (r - e_r) - e_u) \\
&= Ax^* + BF\hat{x}^* - BFe_x + Br - Be_r - Be_u \\
&= Ax^* + BF\hat{x}^* - [\,BF \;\, B \;\, 0 \;\, B\,] \begin{bmatrix} e_x \\ e_r \\ e_y \\ e_u \end{bmatrix} \tag{4.44}
\end{align*}$$

Define

$$A_o = \begin{bmatrix} A & BF \\ -LC & \Phi \end{bmatrix}; \quad B_o = \begin{bmatrix} B \\ B \end{bmatrix}$$

Then,

$$\begin{align*}
\begin{bmatrix} \dot{x}^* \\ \dot{\hat{x}}^* \end{bmatrix} &= A_o \begin{bmatrix} x^* \\ \hat{x}^* \end{bmatrix} + B_o r - \eta E \\
y^* &= C_o \begin{bmatrix} x^* \\ \hat{x}^* \end{bmatrix} \tag{4.45}
\end{align*}$$

where $C_o = [\,C \;\, 0\,], E = [\,e_x \;\, e_r \;\, e_y \;\, e_u\,]^T$, and $\eta = \begin{bmatrix} BF & B & 0 & B \\ \Phi & B & -L & 0 \end{bmatrix}$.

Finally, define the combined state error vector

$$x'[k] = \begin{bmatrix} x[k] - x^*[k] \\ \hat{x}[k] - \hat{x}^*[k] \end{bmatrix}; \quad \Delta y[k] = y[k] - y^*[k] \tag{4.46}$$

Then, combining (4.43), (4.44), (4.45), and (4.46),

$$x'[k+1] = A_o x'[k] + \eta E[k]$$

$$\Delta y[k] = C_o x'[k]$$

The steady-state output error variance is

$$\sigma_{\Delta y[k]} = \operatorname{tr}(\bar{W}_o \eta \eta^T)\sigma^2$$

See [62, Chapter 9] for more details.

4.12 Other Measures

Due to lack of time, I could not include the development of some other measures. I will leave them for future work. However, I have included citations and a short description of them here.

Gevers and Li [62] develop a synthetic measure for controllers that takes into account closed-loop performance. This synthetic measure combines the closed-loop noise gain and the closed-loop sensitivity measures developed in the previous two sections.

I include in Section 5.10 a development of the LQG cost (due to Liu *et al.* [105]) that takes into account roundoff noise.

Rotea and Williamson [152] derive optimal realizations based on either H_2 or H_∞ roundoff noise gain measures, subject to H_2 or H_∞ scaling constraints. The H_∞ noise gain "gives the maximum possible variance when the quantization error has bounded variance but an unknown power spectral density (PSD)" [152]. They also give synthesis algorithms to solve the problem of controller/filter design, optimizing for the chosen noise and scaling norms.

The next chapter will present most of the corresponding realizations that minimize (or maximize) each of these measures.

Chapter 5

Optimizations

Every “optimal” or “suboptimal, but good” realization is optimal or good with respect to a measure (perhaps more than one). The previous chapter discussed many of the currently popular measures and suggests more. This chapter delves into the actual optimal structures for each measure. Organized like the previous one, this chapter first develops optimal filter structures and then discusses controller structures. Within these categories, I have tried to maintain the distinction between the deterministic and stochastic realms, but I cross over the line where it will improve the presentation.

The first section describes some common filter realizations. The first subsection starts with the direct-form structures and their poor FWL performance, and thereby motivates the remaining subsections, which include filter designs that perform significantly better. Many of these filter structures are quite well known and are described in detail in most introductory DSP texts (for eg. [130]. Though not optimal with respect to any particular measure, they still have good overall performance in terms of coefficient quantization sensitivity and roundoff noise gain. Sections 5.2, 5.3, 5.4, and 5.5 each present the transformations that minimize $M_{L_{12}}$, M_{L_2}, M_{pz}, and $M^*_{L_{12}}$ respectively. Sections 5.6 and 5.7 describe the minimum roundoff noise gain and minimum synthetic noise gain realizations. Section 5.8 covers block processing, which can reduce computation and roundoff noise. Block processing has found interesting application in controls problems under the name ‘lifting’. Section 5.9 moves one step up (or back) in the design hierarchy into discrete (or digital) redesign techniques.

The premise is to design a digital filter or controller based on the coefficients of an ideal CT filter or controller. The first subsections cover filter redesign. Again, this area is well understood and much of the material can be found in introductory DSP text books. The later subsections cover compensator discretization and introduce some novel ideas like lifting to take into account the plant model in the discretization process. The subsections also mention direct DT controller design from CT plants. Section 5.10 mentions the LQG design methods of Moroney *et al.* and then describes in detail Liu *et al.*'s 'shortcut' that does direct FWL LQG *design* for a discretized plant. Section 5.11 discusses sparseness in controllers and filters and includes Amit and Shaked's 0,1 algorithm [3] (an extension of Bomar and Hung's work [23]). Finally, Section 5.12 gives references to the optimizations that I did not have a chance to develop in this thesis. I leave them for future work.

As in earlier chapters, one must remember throughout this chapter that digital signal processing problems and solutions are a subset of controls problems and solutions. After all, a digital compensator *is* a digital filter if one ignores the plant. However, a compensator has to satisfy more severe constraints, such as on computational delay. For example, pipelining a digital filter will not affect filter performance since the critical measure is throughput. For a controller, though, latency is much more important, and thus pipelining is not always feasible (see [119] for more details on architectural issues). Let us continue our journey.

5.1 Realization Structures

Researchers have developed many structures that, assuming infinite precision, implement the same transfer function

$$H(z) = \frac{N(z)}{D(z)} = \frac{b_0 + b_1 z + \ldots + b_m z^m}{a_0 + a_1 z + \ldots + a_n z^n}$$

from an input/output standpoint. However, they all have very different performance characteristics with respect to FWL effects. Starting with the simple (and *very* sensitive) direct-form structures, the next few subsections describe several of the more complicated structures as well as the advantages and disadvantages of each.

The state-space form can represent most of the structures. However, to fully capture FWL effects, the state-space notation must incorporate the order of operations. Several authors have suggested alternatives. I will briefly mention some of the ones I came across. Belter and Bass [19] describe notation to capture arbitrary circuit topology. Chan [33] developed a notation that extends the state-space to include a notion of precedence levels. Precedence levels impose an order of operations and also indicate which sets of operations can occur in parallel and which these sets must be executed sequentially. Moroney *et al.* extended Chan's notation to account for compensators. Roberts and Mullis [151] discuss the 'factored' state variable description which is similar to the standard state-space description but also captures the order of operations in the realization. Judging from the dearth of literature, though, none of these notations has been particularly popular, at least in FWL research.

I begin with the simplest structures, which realize the TF as directly as possible.

5.1.1 Direct Forms I and II

The Direct Form (DF) structures are two of the simplest structures. The DF II structure, shown in Figure 5-1, is also called a *canonical* form since it has the minimum number of delays that an *nth*-order system will have. In terms of hardware cost, this structure is the cheapest since the number of multipliers required is minimal (assuming that multipliers dominate the cost function). Nowadays though, in many DSPs and integrated circuits, a multiply may not take any more time or space/money than other arithmetic operations, often nullifying this advantage.

The direct form's biggest disadvantage is its sensitivity to parameter perturbations, especially as the filter order increases [88]. Increasing sampling rate while discretizing a CT filter to design a DT filter also increases sensitivity, setting up conflicting (and somewhat counterintuitive) goals. A higher sampling rate better approximates the original CT filter but also has much poorer FWL performance[1]

[1] In the controls context, a higher sampling reduces aliasing effects and may be required, depending on the CL bandwidth of the controller or filter. Thus, discretizing the CT controller at this higher rate and implementing it in direct form will lead to a poor FWL compensator.

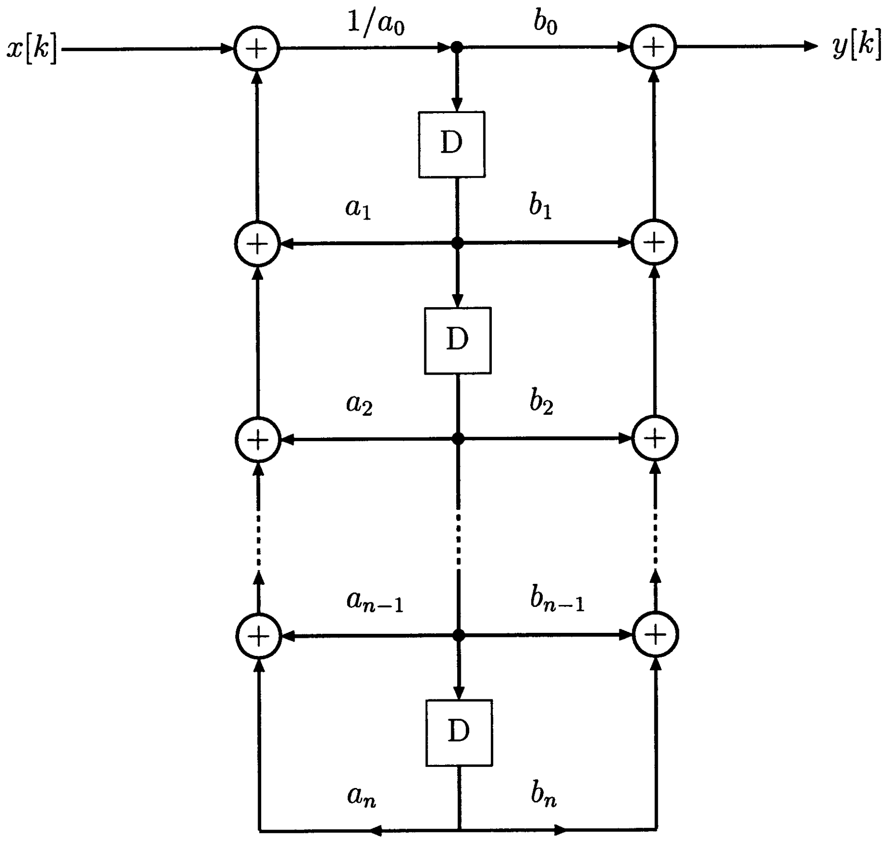


Figure 5-1: Direct Form II.

The DF I structure, shown in Figure 5-2, reverses the order of the two sections that make up the DFII structure. Why would someone choose the DF I structure? Why introduce additional delay blocks? Because the sensitivity/roundoff noise may be lower than for the DF II structures, depending on the noise norm being used [78,79,81].

The transposed DF I and DF II structures also minimize different noise measures. Transposition simply reverses the direction of each signal and exchanges the input and output. A cursory comparison of Figures 5-2 and 5-4 and of Figures 5-1 and 5-3 will clarify transposition.

Since increased sensitivity is a property of any higher-order polynomial, one would expect the same sensitivity problems in high-order finite impulse response (FIR) filters. However, the difference is that generally FIR filters, used for their linear

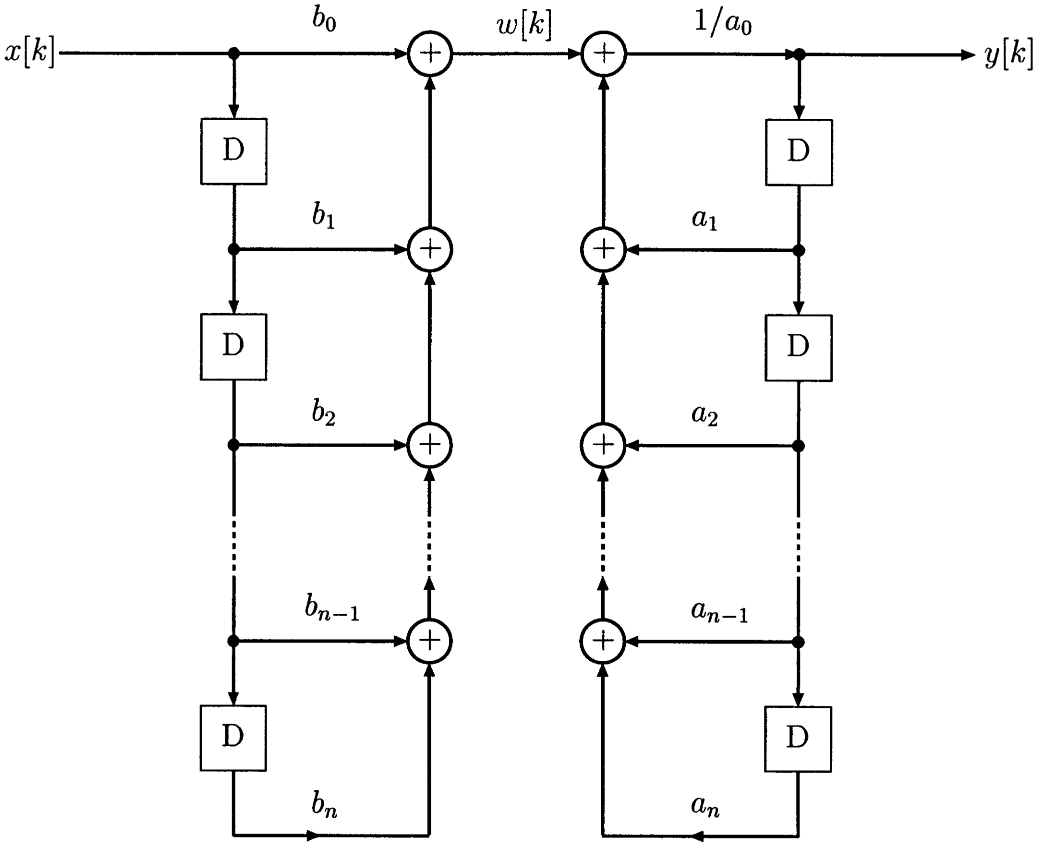


Figure 5-2: Direct Form I.

phase properties, satisfy a symmetry condition on their coefficients:

$$b_i = \pm b_{m-i}$$

When quantized, the coefficients will still satisfy the symmetry constraint. Thus, only the magnitude response of the filter will change. Zeros on the unit circle will stay on the unit circle unless they are perturbed enough to come together and split into reciprocal pairs [81, Chapter 11].

Since the sensitivity and noise gain for higher-order systems for each of these four structures is very large, higher-order systems are usually decomposed into lower-order subsections. The two simplest decompositions are the cascade and parallel decompositions.

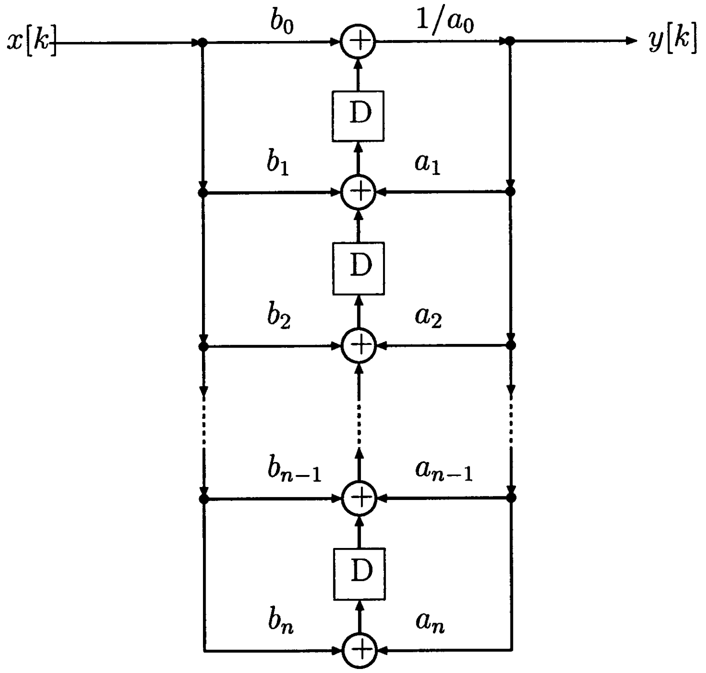


Figure 5-3: Direct Form II Transpose.

5.1.2 Cascade Decomposition

One can decompose a higher-order transfer function into a cascade of first- and second-order sections that can each be implemented in direct form (or in another realization), see Figure 5-5. Each smaller second-order section is significantly less sensitive since it is of much lower order, and also since tightly clustered poles, a cause of high sensitivity, can be put in separate subfilters.

Another advantage of decomposing a system into second-order subsections is that the number of multiply instructions per output sample is linear in the order of the system. A state-space system with low roundoff noise may require on the order of n^2 instructions per output sample while the direct form nth order structure requires only $2n + 1$ instructions. A decomposition usually compromises between the two and provides good (but suboptimal) roundoff noise performance as well as a linear number of operations per output sample.

Note the design variability inherently introduced by this decomposition. For example consider the decomposition,

$$\frac{(s+1)(s^2+1)}{(s+2)(s+3)(s^2+3)} \rightarrow \frac{s^2+1}{s^2+3} \cdot \frac{s+1}{s+2} \cdot \frac{1}{s+3} \tag{5.1}$$

Different pole-zero pairings and different orderings of the subsections change FWL performance, sometimes significantly [81, 104]. Another direction of design freedom is deciding the realization of each individual subsection. One could simply choose a direct form or could opt for any number of different realizations like the lattice or ladder forms (refer to a DSP text for many of these filter structures). Roberts and Mullis [151] also describe second-order block optimal and sectional optimal forms. Due to the simplicity and popular use of second-order decompositions, a large number of works focus on them (see for eg. [2, 9, 11, 12, 22, 82, 104, 171]).

Again, it is important to reiterate that *these decompositions and their associated properties are only relevant in the FWL context.*[2]

Pole-Zero Pairing Problem

In a cascade decomposition, one must select which zeros should be paired with which poles in each subsection. Each different pairing may result in different FWL behavior. The pole-zero pairing problem is to find the pairing that minimizes output noise gain. One could of course use measures other than output noise gain. Currently, there is no known method to analytically solve the pairing (or ordering) problem. Applying the brute force approach of trying all possible combinations leads to a computational problem that grows approximately as $(n!)^2/(n-m)!$ (m = degree of numerator and n = degree of denominator). Some authors have suggested heuristics that significantly reduce the computational burden. Jackson [78, 81] proposed the following:

(i) Pair the poles closest to the unit circle with the zeros closest to the unit circle.

Poles provide gain, which in the context of FWL effects and limited dynamic range is bad, and zeros provide attenuation. Such a pairing then maintains each subsection's

[2] Acutally, some low senstivity FWL filter designs, such as wave digital filters (WDF) originate from low sensitivity analog filter designs.

gain as close to unity as possible.

Ordering Problem

FWL effects remove the commutativity property of cascaded systems. Thus, a reordering of the subsections can result in significantly different FWL performance. Jackson's rule of thumb for ordering is

(ii) Order the sections such that the most peaked sections (usually the ones with poles closest to the unit circle) appear last.

The exact rule for ordering, however, depends on the noise gain measure used. One has the choice to use an H_2 or H_∞ measure ([81, 152]).

The MATLAB commands `zp2sos` and `ss2sos` use both these rules to produce second-order sections from pole-zero or state-space descriptions, respectively.

Liu and Peled in [104] address both problems with a heuristic that partially uses brute force. They suggest the following optimization algorithm:

(i) Generate a random ordering of the zeros and the poles.

(ii) Perform a local optimization by keeping the zero ordering fixed while interchanging all possible pairs of pole sections. The local optimum from this start is the pairing that generates the minimum output roundoff error E^2. This step requires $(n(n-1)/2+1)$ evaluations of the output error equation.

(iii) Repeat steps (i) and (ii) M times where M is an arbitrary number.

(iv) The best of the M "locally optimal" assignments in step 2 is taken as the "near optimal" assignment. Thus, the total number of output error evaluations is approximately $Mn^2/2$.

They show that most of the time, this heuristic algorithm determines an arrangement with almost the same performance as that of the optimal one as determined by dynamic programming.

5.1.3 Parallel Decomposition

A partial fraction decomposition of an *nth* order system will also result in first- and second-order subsections, though these will be connected in parallel (Figure 5-6). This decomposition uniquely determines pole-zero pairing and does not require any choices by the designer.

Problems with a Parallel Decomposition

Unfortunately, computing a partial fraction decomposition is a numerically unstable problem. With tightly clustered poles, a small change in a coefficient (such as due to roundoff error) can cause arbitrarily large changes in the resulting decomposition. Furthermore, a parallel decomposition into second-order sections is *not* always possible. In fact, it is only possible with *distinct* poles. Taking these two considerations into account, one might be better off decomposing a system into paralleled higher-order sections, each of which includes identical poles and poles close to each other in a cascade. A particular 'paralleled cascade' decomposition may then read something like

$$H(z) = H_n(z)H_{n-1}(z) + \ldots + H_6(z)H_5(z)H_4(z)H_3(z) + H_2(z) + H_1(z)$$

where each of the $H_i(z)$ is a second-order section; $H_3(z), H_4(z), H_5(z)$, and $H_6(z)$ have poles that are identical or very close to each other, and so the sections are cascaded.

Surprisingly, practically none of the digital signal processing texts that I referred to mentioned the numerical instability, and most also describe the parallel decomposition (into second-order sections) as a possibility for *every* $H(z)$. The only references that did mention the numerical instabilities were the help command for `residue` in MATLAB and Hanselmann's survey [68] on digital controllers.

Interestingly, [62] points out that if one considers the state transition matrix of the overall system, a parallel decomposition results in a block diagonal form while a cascade decomposition results in a block triangular form, both of which are in Schur form. However, this Schur form usually does not belong to the optimal (with respect to sensitivity or roundoff noise measures) realization set. Going the other way, an algorithm to transform an nth order Schur-form optimal realization into a cascade of second-order sections is not known either.

Another possible decomposition for state-space systems is the Jordan decomposition. However, computing the Jordan decomposition is also a numerically unstable problem. See [139,179] for excellent discussions on numerical aspects of linear systems and controls algorithms.

Differences between a parallel and cascade decomposition

Apart from the fact that every TF has a cascade decomposition and not a parallel one, there are some other differences between the two. Jackson [81] points out that in a cascade decomposition, the coefficients in each subsection's denominator depend only on the coefficients in the denominator of the original TF and similarly with the numerator. In a parallel decomposition though, the decomposed sections' numerator coefficients depend on the coefficients of *both* the numerator and denominator of the original TF. Since the zeros significantly affect magnitude characteristics, the design should be rechecked after a parallel decomposition, if attenuation requirements are stringent, to ensure that the quantized transfer function still meets the specifications [81].

5.1.4 Other Filter Structures

Several researchers have proposed different filter realizations that have low sensitivity and good roundoff noise properties. Some of the more popular ones include:

- Lattice structures ([102,113,114])
- Wave Digital Filters (WDF) – which include the lattice and ladder structures and other more general structures (Fettweis has an excellent survey article [52] and Rao and Kailath discuss VLSI implementation of these structures [147]).
- The sectional and block optimal forms (see Sections 5.6.1 and 5.6.2 respectively)

- Block processing and other interleaved/pipelined architectures for high throughput rates (see Section 5.8).

Bomar [22] developed several new structures, starting with the following basic idea: Take a second-order (symbolic) state-space structure, compute its input/output description, and match the coefficients with a transfer function. A fully parametrized state-space structure has eight coefficients while the transfer function only has four variables. Thus, one can impose upto four more constraints. Two important ones are the scaling constraints on the controllability Gramian, W_c: $[W_c]_{ii} = 1$ for $i = 1, 2$ (see Section 2.3). To minimize the trace of the observability Gramian, W_c (defined as the roundoff noise gain, G in Section 4.5), or some other measure, the remaining two parameters can be chosen freely and are what lead to a variety of different final structures. For example, choosing $a_{11} = a_{22}$ and $b_1c_1 = b_2c_2$ yields the minimum-noise structure. Using $a_{11} = 0$ and $b_1 = 0$ yields the familiar canonical form. He develops a few more structures and also compares the performance of several of these structures.

Other structures have been proposed in the literature and including them all could fill up several more pages. Unfortunately, I will have to leave those for future work.

The usefulness of many of these structures in the controls context has not specifically been investigated. Obviously, their desirable properties as filters carry over when they are used for digital controller implementations. However, any particular advantage or optimization with respect to controls is unknown (see the discussion on block processing in Section 5.8 for some specific suggestions for areas of investigation).

5.1.5 Non-Minimal Realizations

One could reasonably ask why should the search for the optimal structure leave out non-minimal realizations? After all, “hiding” noise in the unobservable and/or uncontrollable modes may reduce output noise. Indeed, some researchers have followed up on this. Beex and Debrunner in [18] examine the influence of introducing (judiciously placed) pole-zero cancellations while Tokaji and Barnes in [174] examine the same question for roundoff noise from the state-space view.

Beex and Debrunner show that for low-order systems, one can add pole/zero cancellations to reduce sensitivity while maintaining a low-computation Direct Form II structure. However, for higher-order systems (10th-order all-pole low-pass Butterworth in their example), the sensitivity of the Direct Form, even after introducing some pole-zero cancellations, is still orders of magnitude above that of the minimum roundoff noise optimal form. Their example also shows another advantage of the optimal form (which is introduced in Section 5.6): its sensitivity stays constant as the filter bandwidth decreases, while that of the Direct Form increases without bound.

Note that the scattered look-ahead (SLA) filters and some of the other pipelining methods mentioned in Section 5.8 use the cancellation of additional poles and zeros to pipeline IIR filters. Comparing the roundoff noise analyses for the SLA filters with the techniques used by Beex and Debrunner may also point to an interesting overlap.

I have grouped Tokaji and Barnes' work with the material on roundoff noise in Section 5.6.3.

5.2 Minimum $M_{L_{12}}$ Sensitivity Realization

The $M_{L_{12}}$ measure, given in Section 4.1, is

$$M_{L_{12}} \triangleq \left\| \frac{\partial H(z)}{\partial A} \right\|_1^2 + \left\| \frac{\partial H(z)}{\partial B} \right\|_2^2 + \left\| \frac{\partial H(z)}{\partial C} \right\|_2^2$$

with a SISO upper bound

$$\bar{M}_{L_{12}} = \operatorname{tr}(W_c)\operatorname{tr}(W_o) + \operatorname{tr}(W_o) + \operatorname{tr} W_c$$

Thiele [172] first proved the following:

$$\operatorname{tr}(W_c)\operatorname{tr}(W_o) \geq \left(\sum_{i=1}^{n} \nu_i \right)^2 \tag{5.2}$$

Each of the ν_i is termed a second-order mode of the state-space system. A second-order mode, also called a Hankel singular value, is an eigenvalue of the product $W_c W_o$. Note that these eigenvalues are invariant under similarity transformations on the system (because $W_c W_o$ is also transformed by similarity). Thus, the second-order modes characterize the system, and the similarity transformations that make $\mathrm{tr}\,(W_c)\mathrm{tr}\,(W_o)$ achieve equality in (5.2) are candidates for being optimal transformations. One can also easily show that

$$\mathrm{tr}\,(W_c) + \mathrm{tr}\,(W_o) \geq 2\sum_{i=1}^{n} \nu_i \tag{5.3}$$

by starting with the inequality

$$\begin{aligned} \left(\sqrt{\mathrm{tr}\,(W_c)} - \sqrt{\mathrm{tr}\,(W_o)}\right)^2 &\geq 0 \\ \mathrm{tr}\,(W_c) + \mathrm{tr}\,(W_o) &\geq 2\sqrt{\mathrm{tr}\,(W_c)\mathrm{tr}\,(W_o)} \\ &= 2\sqrt{\left(\sum_{i=1}^{n} \nu_i\right)^2} \\ &= 2\sum_{i=1}^{n} \nu_i \end{aligned}$$

Thiele [172] proved that equality in (5.2) holds if and only if $W_o = \alpha W_c$ for a scalar $\alpha \neq 0$, while equality in (5.3) holds if and only if $W_o = W_c$. Thus, the realizations that minimize the upper bound are those that have equal observability and controllability Gramians. These realizations are called the internally balanced realizations. In [173], he also proved that the internally balanced realizations minimize not only the upper bound $\bar{M}_{L_{12}}$ but in fact minimize $M_{L_{12}}$ itself. $M_{L_{12}}$ is the equal to $\bar{M}_{L_{12}}$ for the optimal transformations. See [62, 172] for a characterization of the optimal similarity transformations that will change any system to an internally balanced realization. In MATLAB , the commands `balreal` and `ssbal` produce internally balanced realizations.

Optimizing a MIMO realization for $M_{L_{12}}$

The upper bound, $\bar{M}_{L_{12}}$, in the MIMO case is, from (4.17), given by

$$\bar{M}_{L_{12}} = \mathrm{tr}\,(W_c)\,\mathrm{tr}\,(W_o) + m\,\mathrm{tr}\,(W_o) + p\,\mathrm{tr}\,(W_c)$$

To determine the realization that will minimize this measure, consider each term. From the SISO case, we know that

$$\mathrm{tr}\,(W_c)\mathrm{tr}\,(W_o) \geq \left(\sum_{i=1}^{n} \nu_i\right)^2 \tag{5.4}$$

with equality if and only if

$$W_c = \alpha W_o \tag{5.5}$$

for some scalar $\alpha \neq 0$.

Thus, (5.4) bounds the first term. To bound the second and third term, [109] use the following argument:

$$\begin{aligned}\left[\sqrt{p\mathrm{tr}\, W_c} - \sqrt{m\mathrm{tr}\, W_o}\right]^2 &\geq 0 \\ m\mathrm{tr}\, W_o + p\mathrm{tr}\, W_c &\geq 2\sqrt{pm(\mathrm{tr}\,(W_c)\mathrm{tr}\,(W_o))} \\ m\mathrm{tr}\, W_o + p\mathrm{tr}\, W_c &\geq 2\sqrt{pm}\left(\sum_{i=1}^{n}\nu_i\right)\end{aligned}$$

with equality if and only if $W_c = (m/p)W_o$.

Satisfying the condition for minimality for the second and third term automatically satisfies (5.5). Thus, the upper bound has the following minimum:

$$\bar{M}_{L_{12}} = \left(\sum_{i=1}^{n}\nu_i\right)^2 + 2\sqrt{pm}\left(\sum_{i=1}^{n}\nu_i\right)$$

Amazingly, the lower bound is no more complicated than a scaled version of the minimum in the SISO case.

To find the realization that achieves this minimum, first find the balanced realization of $H(z)$. For the balanced realization, $W_c = W_o = \Sigma$. Then, apply the transformation $T = \sqrt[4]{p/m}I$. The resulting W_c, W_o will be $W_c = \sqrt{m/p}\Sigma$ and $W_o = \sqrt{p/m}\Sigma$ with $W_c = \frac{m}{p}W_o$, achieving the minimum.

5.3 Minimum M_{L_2} Sensitivity Realization

The M_{L_2} measure, derived in Section 4.2, is

$$M_{L_2} \triangleq \left\|\frac{\partial H(z)}{\partial A}\right\|_2^2 + \left\|\frac{\partial H(z)}{\partial B}\right\|_2^2 + \left\|\frac{\partial H(z)}{\partial C}\right\|_2^2$$

The first term reduced to

$$\begin{aligned}\left\| \frac{\partial H(z)}{\partial A} \right\|_2^2 &= \frac{1}{2\pi}\int_0^{2\pi} \operatorname{tr}\left(\sum_{k=0}^{\infty} h^T[k]h[k] \right) \, d\omega \\ &= \operatorname{tr}\left(\sum_{k=0}^{\infty} h^T[k]h[k] \right) \\ &\triangleq \operatorname{tr}\left(W_A\right)\end{aligned}$$

where $h[k]$ is defined in (4.18). M_{L_2} then simplifies to

$$M_{L_2} = \operatorname{tr}\left(W_A\right) + \operatorname{tr}\left(W_o\right) + \operatorname{tr}\left(W_c\right)$$

To find the measure's minimum and the corresponding set of optimal transformations, we must first study how the measure changes with similarity transformations. Using the definitions of W_c and W_o, we can easily compute $W_c \rightarrow T^{-1}W_cT^{-T}$ and $W_o \rightarrow T^TW_oT$. Also, using the definition of $h[k]$ in (4.18), W_A changes to

$$W_A = \sum_{k=0}^{\infty} T^T h_0[k] T^{-T} T^{-1} h_0^T[k] T$$

where the subscript $_0$ indicates the matrix defined by the original system (A, B, C, D), so

$$M_{L_2} = \operatorname{tr}\left(\sum_{k=0}^{\infty} T^T h_0[k] T^{-T} T^{-1} h_0^T[k] T \right) + \operatorname{tr}\left(T^T W_o^0 T\right) + \operatorname{tr}\left(T^{-1} W_c^0 T^{-T}\right)$$

where W_c^0 and W_o^0 are the Gramians of the original system. Reordering the matrices and defininining $P = TT^T$,

$$M_{L_2} = R(P) \triangleq \operatorname{tr}\left(\sum_{k=0}^{\infty} P_0^h[k] P^{-1} h_0^T[k] \right) + \operatorname{tr}\left(PW_o^0\right) + \operatorname{tr}\left(P^{-1}W_c^0\right).$$

Gevers and Li prove that $R(P)$ has a globally unique minimum, and it is achieved only by a positive definite P [62, Chapter 5]. Since the form they present M_{L_2} in leads to no obvious analytical solution, they compute the derivative of $R(P)$ with respect to P and then use a gradient descent equation of the form

$$P[k+1] = P[k] - \mu \frac{\partial R(P)}{\partial P} \Big|_{P=P[k]}$$

where μ is a positive step size. The descent is guaranteed to converge to the global minimum (since it is unique) with a small enough step size μ and a positive definite initial condition. There are no hard-and-fast rules for selecting a proper step size [62]. When doing simulations, I used a step size that modified iteself as the search progressed. If after 10 or 15 update steps the solution is still positive definite, it means the search is going in the right direction, and thus, I attempt to double the step size. If the resulting P at any point is negative definite, the algorithm backs up, reduces the step size in half, and continues. The convergence condition is that $\dfrac{\partial R(P)}{\partial P}$ be small. The convergence is significantly faster if started with a 'good' initial realization, such as the $M_{L_{12}}$ optimal realization (which can be computed analytically).

Once P_{opt} is determined, the optimal transformation T_{opt} is any square root

$$T_{opt} = X\Lambda^{1/2}X^T$$

where X is the matrix of right eigenvectors of P and Λ is the diagonal matrix of eigenvalues of P. Since P is symmetric, it always has a full set of orthogonal eigenvectors, and since it is positive definite, all entries of Λ are greater than zero.

Note that the development in this chapter ignored the l_2 scaling constraint. The next section shows how to incorporate the constraint using Lagrange multipliers.

5.4 Minimum M_{pz} Sensitivity Realization

Recall that M_{pz} is defined as

$$\begin{aligned} M_{pz} &\triangleq \sum_{i=1}^{n} w_{\lambda_i} \left\| \frac{\partial \lambda_i}{\partial A} \right\|_F^2 + w_{v_i} \left(\left\| \frac{\partial v_i}{\partial A} \right\|_F^2 + \left\| \frac{\partial v_i}{\partial B} \right\|_F^2 + \left\| \frac{\partial v_i}{\partial C} \right\|_F^2 + \left\| \frac{\partial v_i}{\partial D} \right\|_F^2 \right) \\ &= \sum_{i=1}^{2n} \operatorname{tr}\left(H_i H_i^H\right) + \operatorname{tr}\left(M_y\right) + \operatorname{tr}\left(M_x\right) + c \end{aligned}$$

in Section 4.3.

Including the changes due to the similarity transformation T on the eigenvectors of A and Z,

$$\begin{aligned} M_{pz} &= \sum_{i=1}^{2n} \operatorname{tr}(PH_kP^{-1}H_k^H) + \operatorname{tr}(PM_y) + \operatorname{tr}(P^{-1}M_x) + c \\ &\triangleq R(P) \end{aligned}$$

where $P = TT^T$ and

$$\begin{aligned} H_k &= w_{\lambda_k}^{1/2} y_{p_k}^0 x_{p_k}^{0H}, k = 1, ..., n \\ &= w_{v_k}^{1/2} y_{z_{k-n}}^0 x_{z_{k-n}}^{0H}, k = n+1, ..., 2n \\ M_y &= \sum_{k=1}^{n} w_{v_k} \alpha_k^2 y_{z_k}^0 y_{z_k}^{0H} \\ &= Y_z^0 diag(w_{v_1}\alpha_1^2, ..., w_{v_n}\alpha_n^2) Y_z^{0H} \\ M_x &= \sum_{k=1}^{n} w_{v_k} \beta_k^2 x_{z_k}^0 x_{z_k}^{0H} \\ &= X_z^0 diag(w_{v_1}\beta_1^2, ..., w_{v_n}\beta_n^2) X_z^{0H} \\ c &= \sum_{k=1}^{n} w_{v_k} \alpha_k^2 \beta_k^2 \end{aligned}$$

where α_k and β_k are coordinate independent and defined in (4.29) and (4.30) and repeated here:

$$\begin{aligned} \alpha_k^2 &\triangleq \left|d^{-1} x_{z_k}^H C^T\right|^2 = \left|d^{-1} C x_{z_k}\right|^2 \\ \beta_k^2 &\triangleq \left|d^{-1} B^T y_{z_k}\right|^2 \end{aligned}$$

The superscript ^{0} indicates the left and right eigenvectors corresponding to the original system (A, B, C, D).

Just as in the case for M_{L_2}, solving for P analytically seems impossible, and Gevers and Li turn to gradient descent after proving that a unique, positive definite P is guaranteed to exist and achieves the globally unique minimum of the measure. With the following expression for the partial of $R(P)$,

$$\frac{\partial R(P)}{\partial P} = \sum_{k=1}^{2n} H_k P^{-1} H_k^H - P^{-1} H_k^H P H_k P^{-1} + M_y - P^{-1} M_x P^{-1}$$

they apply the gradient equation,

$$P(k+1) = P(k) - \mu \frac{\partial R(P)}{\partial P} \,|_{P=P(k)}$$

One is guaranteed to reach the minimum (as long as μ is small enough) since there are no local maxima or minima.

Now, to include the l_2 constraint, one needs to use Lagrange multipliers

$$\begin{aligned} L(P,\lambda) &= R(P) + \lambda[\operatorname{tr}(W_c^0 P^{-1}) - n] \\ \Rightarrow \frac{\partial L}{\partial P} &= \sum_{k=1}^{2n} H_k P^{-1} H_k^H - P^{-1} H_k^H P H_k P^{-1} + M_y - P^{-1} M_x P^{-1} - \lambda P^{-1} W_c^0 P^{-1} = 0 \\ \frac{\partial L}{\partial \lambda} &= \operatorname{tr}[W_c^0 P^{-1}] - n = 0 \end{aligned}$$

and then use the simultaneous gradient descent equations

$$\begin{aligned} P(k+1) &= P(k) - \mu_1 \frac{\partial L(P,\lambda)}{\partial P} \Big|_{\substack{P=P(k) \\ \lambda=\lambda(k)}} \\ \lambda(k+1) &= \lambda(k) - \mu_2 \frac{\partial L(P,\lambda)}{\partial \lambda} \Big|_{\substack{P=P(k) \\ \lambda=\lambda(k)}} \end{aligned}$$

5.5 Minimum Frequency-Weighted $M_{L_{12}}$ Realizations

The frequency weighted measure, $M_{L_{12}}^*$ from Section 4.4, is

$$\begin{aligned} M_{L_{12}}^* \triangleq & \left\| W_A(z) \frac{\partial H(z)}{\partial A} \right\|_2^2 \\ & + \|W_B(z)G(z)\|_2^2 + \|W_C(z)F(z)\|_2^2 \end{aligned}$$

with an upper bound

$$\begin{aligned} \bar{M}_{L_{12}}^* \triangleq & \|W_1(z)G(z)\|_2^2 \, \|W_2(z)F(z)\|_2^2 \\ & + \|W_B(z)G(z)\|_2^2 + \|W_C(z)F(z)\|_2^2 \end{aligned}$$

where the weighting function for the derivative with respect to A is $W_A(z) = W_1(z)W_2(z)$ (see Section 4.4 for more details about the measure).

Applying a transformation T to the system, the frequency weighted Gramians change as their corresponding Gramians (observability or controllability) do:

$$K_{o1} \rightarrow T^T K_{o1} T; \quad K_{c2} \rightarrow T^{-1} K_{c2} T^{-T}$$

$$K_{oB} \rightarrow T^T K_{oB} T; \quad K_{cC} \rightarrow T^{-1} K_{cC} T^{-T}$$

resulting in the following upper bound minimization problem:

$$\min_{\{T \mid \det T \neq 0\}} \bar{M}^*_{L12} = \operatorname{tr}(T^T K_{o1} T)\operatorname{tr}(T^{-1} K_{c2} T^{-T}) + \operatorname{tr}(T^T K_{oB} T) + \operatorname{tr}(T^{-1} K_{cC} T^{-T})$$

Since $\operatorname{tr}(BA) = \operatorname{tr}(AB)$, the above reduces to

$$R(P) \triangleq \bar{M}^*_{L12} = \operatorname{tr}(K_{o1} P)\operatorname{tr}(K_{c2} P^{-1}) + \operatorname{tr}(K_{oB}) + \operatorname{tr}(K_{cC} P^{-1})$$

where, as before, $P \triangleq TT^T$. Gevers and Li once more use the same approach: prove that there exists a unique positive definite P that achieves the upper bound with equality, and then after computing $\frac{\partial R(P)}{\partial P}$, they apply a gradient descent to find P_{opt}.

The optimal solution above minimizes the upper bound. In the special case that $W_1(z) = W_2(z)$, the optimal solution will minimize the M^*_{L12} measure itself.

This idea of using gradient descent is a powerful one, but to use it successfully, one must first prove the existence and uniqueness of a global minimum. Gevers and Li use it several times throughout their text [62]. If the minimum is not unique though, a gradient descent may get caught in local maxima and minima, and one must be more careful with regards to what search technique is used.

5.6 Minimum Roundoff Noise Structures

Roberts and Mullis, with their 1976 publications, and Hwang in 1977 significantly altered the approach to roundoff noise and sensitivity minimization. They used the state-space setting instead of transfer functions which had dominated FWL analysis until then. Their analyses also clearly depict the connection between scaling and roundoff noise, and lead into the development of the transformations that minimize roundoff noise.

I will follow the development in [151] to show how the controllability and observability Gramians change under similarity transformations, since these two matrices determine scaling and roundoff noise properties.

The observability Gramian, W_o, is the solution of the Lyapunov equation $W_o = A^T W_o A + C^T C$.

$$W_o = \sum_{i=0}^{\infty} (A^T)^i C^T C A^i$$

Then, under the similarity transformation T, $(A, B, C) \rightarrow (T^{-1}AT, T^{-1}B, CT)$ and W_o will change to $T^T W_o T$. In Section 4.1, I developed a similar expression for W_c, see (4.14). W_c changes to $T^{-1} W_c T^{-T}$.

Now, consider the diagonal transformation determined by the l_2 scaling rule: $[T]_{ii} = \alpha\sqrt{[W_c]_{ii}}$. Under this transformation, the diagonal elements of W_c change to $\frac{1}{\alpha^2}$ as expected and desired. However, the effect on the diagonal elements of W_o is $[W_o']_{ii} = T^T [W_o]_{ii} T = \alpha^2 [W_c]_{ii} [W_o]_{ii}$. The total roundoff noise gain, $\mathrm{tr}\,(W_o')$, for the *unscaled* filter will change to

$$\sigma^2_{total} = \alpha^2 \sigma^2 \sum_{i=1}^{n} [W_c]_{ii} [W_o]_{ii} \tag{5.6}$$

where σ^2 is the variance of one noise source. As α is increased, the probability of overflow will decrease, but the output noise variance will increase. Thus, one should increase α only as much as is necessary.

To investigate minimum roundoff noise structures, consider how the product $W_c W_o$ changes under a similarity transformation T:

$$W_c \rightarrow T^{-1} W_c T^{-T};\, W_o \rightarrow T^T W_o T$$

$$W_c W_o \rightarrow T^{-1} W_c W_o T \tag{5.7}$$

The eigenvalues of $W_c W_o$, the second order modes of the system, are thus also invariant under similarity transformations, as noted earlier.

One can generate minimal noise structures by either allocating an equal number of bits to all state registers (the equal wordlength (EWL) case) or by optimally allocating bits to state registers (the optimal wordlength (OWL) case), i.e. allocate more bits to the 'more important' states. In both cases, we have a fixed number of total bits, nB to allocate. Thus, $\sum_{i=1}^{n} B_i = nB$ where B_i is the number of bits allocated to the ith state variable's register.

Optimal Wordlength (OWL) case

The ith state variable (or register) has a quantization step 2^{-B_i} with a resulting variance $\sigma_i^2 = \frac{2^{-2B_i}}{12}$. Substituting into (5.6),

$$\sigma_{total}^2 = \frac{\alpha^2}{12}\sum_{i=1}^{n}\frac{[W_c]_{ii}[W_o]_{ii}}{(2^{B_i})^2} \tag{5.8}$$

Using the arithmetic-geometric mean inequality,

$$\frac{1}{n}\sum_{i=1}^{n}\frac{[W_o]_{ii}\,[W_c]_{ii}}{2^{2B_i}} \geq \left[\prod_{i=1}^{n}\frac{[W_o]_{ii}\,[W_c]_{ii}}{2^{2B_i}}\right]^{1/n} \tag{5.9}$$

To optimize the choice of B_i, we choose them so that equality is achieved, i.e. all terms are equal. Let

$$c = \frac{[W_o]_{ii}\,[W_c]_{ii}}{2^{2B_i}}, \qquad\qquad i = 1, 2, ..., n \tag{5.10}$$

and choose B_i such that $\sum\limits_{i=1}^{n} B_i = nB$, resulting in

$$\begin{aligned} c^n &= \frac{\prod_{i=1} n \frac{[W_o]_{ii}[W_c]_{ii}}{2^{2B_i}}}{2^{\left(2\sum\limits_{i=1}^{n} B_i\right)}} \\ &= \frac{\prod_{i=1} n \frac{[W_o]_{ii}[W_c]_{ii}}{2^{2B_i}}}{2^{2nB}} \end{aligned}$$

Taking logarithms and substituting back with the original definition of c in (5.10),

$$B_i = B + \frac{1}{2}\log_2([W_c]_{ii}\,[W_o]_{ii}) - \frac{1}{2n}\sum_{j=1}^{n}\log_2([W_c]_{jj}\,[W_o]_{jj}) \tag{5.11}$$

which achieves equality in (5.9) and gives an output noise of

$$\sigma_{total,OWL}^2 = \left[\frac{n}{12}\left(\frac{\alpha}{2^B}\right)^2\right]\left[\prod_{i=1}^{n}[W_c]_{ii}\,[W_o]_{ii}\right]^{1/n} \tag{5.12}$$

Note that (5.11) typically does not give integral values for the B_i, and (5.12) says nothing about how to search for a similarity transformation to achieve the minimum value for the right hand side. Mullis and Roberts use Hadamard's inequality to minimize (5.12), the geometric mean of the diagonal elements of W_c and W_o. They derive the condition

$$\sigma^2_{total,OWL} = \left[\frac{n}{12}\left(\frac{\alpha}{2^B}\right)^2\right] \frac{M_g^2}{[e(W_c)e(W_o)]^{2/n}} \tag{5.13}$$

where

$$M_g = [\det(W_cW_o)]^{1/2n} = \left[\prod_{i=1}^{n} \nu_i^2\right]^{1/2n} = \left[\prod_{i=1}^{n} \nu_i\right]^{1/n}$$

and where $e(A)$ is the scalar defined as

$$0 \leq e(A) = \left[\frac{\det A}{\prod_{i=1}^{n}[A]_{ii}}\right]^{1/2} \leq 1$$

Equation (5.13) is minimized when $e(W_c) = e(W_o) = 1$. By Hadamard's inequality, this will be true if and only if both W_c and W_o are simultaneously diagonal (see Section C.1 in appendix C for a more detailed derivation), in which case

$$\sigma^2_{total,OWL} = \left[\frac{n}{12}\left(\frac{\alpha}{2^B}\right)^2\right] M_g^2$$

Mullis and Roberts leave off here, noting that the simultaneous diagonalization of two symmetric matrices is a common linear algebra problem. They more specifically address the case of equal wordlength (EWL) optimization.

Equal Wordlength (EWL) case

If all $B_i = B$, then (5.8) reduces to

$$\sigma^2_{total} = \left[\frac{n}{12}\left(\frac{\alpha}{2^B}\right)^2\right] \left[\frac{1}{n}\sum_{i=1}^{n}[W_c]_{ii}[W_o]_{ii}\right]$$

The following result (due to [122]) leads to the minimization of this problem:

$$\frac{1}{n}\sum_{i=1}^{n}[W_c]_{ii}[W_o]_{ii} \geq M_a^2, \qquad M_a = \frac{1}{n}\sum_{i=1}^{n}\nu_i$$

with equality if and only if

$$W_c = DW_oD, \qquad D = \text{diag}\{d_1^2, d_2^2, ..., d_n^2\}$$

$$[W_c]_{ii}\,[W_o]_{ii} = [W_c]_{jj}\,[W_o]_{jj}\,, \qquad i, j = 1, 2, ..., n$$

Thus,

$$e(W_c) = e(W_o) = \left(\frac{M_g}{M_a}\right)^n$$

$$\sigma^2_{total,EWL} = \left[\frac{n}{12}\left(\frac{\alpha}{2^B}\right)^2\right] M_a^2 \tag{5.14}$$

M_a^2 is the noise gain of the EWL filter. The ratio of M_g^2/M_a^2 is the advantage one can expect to gain by distributing bits optimally among the registers, with the largest number for the ith state variable if ν_i is the largest second-order mode. Williamson in [185] also discusses optimal (unequal) distribution of bits. He also points out that reducing the number of bits assigned to a state variable is equivalent to doing *partial* order reduction. Thus, this method offers more flexibility than other order reduction methods which produce an all-or-none answer. I will revisit this idea in Section 5.10 in the context of LQG controller design.

The transformation that brings an abritrary realization (A, B, C, D) to a realization that achieves the minimum noise (5.14) was originally constructed by Hwang [73] and is given in [62] as:

$$T_{opt} = T_0 U X V^T$$

where

(i) T_0 is a square root of W_c such that $T_0 T_0^T = W_c$,

(ii) $X = diag(x_1, ..., x_n)$ where $x_i = \left(\frac{\sum_{k=1}^{n} \nu_k}{n\nu_i}\right)^{1/2}$,

(iii) U is an orthogonal matrix such that $U^T(T_0^T W_o T_0)U$ is diagonal, and

(iv) V is an orthogonal matrix such that all the diagonal elements of $VX^{-2}V^T$ are equal to one.

For the proof, see [73], and for details on computing the optimal transformation, see [73] or [62].

5.6.1 Sectional Optimal Structures

Noting that most minimum roundoff noise structures are fully parametrized (i.e. have no 0 or ±1 coefficients in the system matrices), Mullis and Roberts suggest a 'sectional' optimal decomposition, a second-order cascade decomposition where each subsection is realized as a minimum roundoff noise structure. The overall cascade will then require about $4n$ multiplies per output sample, a significant improvement over the n^2 multiplies required for a fully parametrized nth order state-space structure. This advantage becomes larger as filter order increases.

Each subsection $H_i(z)$ of the cascade is scaled assuming a white noise input and then transformed into a minimum noise structure. [82] shows that sectional optimal second-order filters are equivalent to minimum noise filters when the following conditions are imposed:

$$a_{11} = a_{22}$$

$$b_1c_1 = b_2c_2$$

and the filter is l_2 scaled.

The importance of the pairing problem (discussed in Section 5.1.2) decreases significantly if each section is realized with an optimal (i.e. minimal noise) realization. The number of multipliers required increases to about $4n$ from the approximately $2.5n$ multipliers required for a cascade decomposition where each section is realized in direct-form. However, by applying scaling one can reduce this number further to $3n$. Bomar in [22] shows several different structures that have near optimal roundoff noise performance and use only seven multipliers per second-order section. The design equations for doing sectional optimal design are given in [151, Chapter 9], along with several design examples.

Note that scaling is done assuming a white noise input for each section; however, downstream sections actually receive colored noise. Thus, the performance of the overall cascade is suboptimal.

5.6.2 Block-Optimal Structures

A block-optimal realization of a cascade of N second-order sections minimizes the roundoff noise gain of each $H_i(z)$ so that the output roundoff noise of the overall cascade is minimized. Block optimal structures differ from sectional optimal structures in the l_2 scaling used. The scaling in a block optimal structure is done taking into account the fact that downstream filters will receive colored noise as input instead of white noise. The design equations are given in [151, Chapter 9].

5.6.3 Non-Minimal Realizations

Tokaji and Barnes [174] prove the following (rather remarkable and surprisingly simple) result: Over the set of non-minimal realizations of dimension m, the minimal attainable noise gain (under an l_2 scaling constraint) is

$$(G_m)_{min} = \frac{n}{m}(G_n)_{min}$$

where n is the dimension of the minimal realization and $(G_n)_{min} = \frac{1}{n}(\sum_{i=1}^{n}\nu_i)^2$ as given earlier in equation (5.14). They go on to point out the impracticality of reducing roundoff noise gain by using non-minimal realizations. To reduce G by 1/2, one would have to double the dimension of the state-space realization, whereas adding 1 bit to the wordlength will decrease G by a factor of 4.

Finally, their result has one other interesting consequence: Minimal noise gain will be attained if and only if all uncontrollable modes are also unobservable. See [174] for details.

5.7 Minimum Synthetic Noise Gain Realizations

The synthetic noise gain measure presented in Section 4.7 combines the effects of coefficient quantization and roundoff noise into one measure. The measure ((4.35)) again is

$$\begin{aligned} G_T &\triangleq \operatorname{tr}(W_A + W_o + W_c) + \frac{\sigma_s^2}{\sigma_c^2}\operatorname{tr}(W_o) \\ &= M_{L_2} + \rho^2 G \end{aligned}$$

where G is the roundoff noise gain measure.

Using the earlier expressions from Section 5.3 for the M_{L_2} minimization problem and from Section 5.6 for the roundoff noise gain term,

$$R(P) \triangleq G_T(T) = \operatorname{tr}\left[\sum_{i=0}^{\infty} P h_0(i) P^{-1} h_0^T(i)\right] + \operatorname{tr}(P M_o^0)$$

where $P = TT^T$ and $M_o^0 \triangleq (1+\rho^2)W_o^0$.

We now have a constrained minimization problem, namely that of minimizing $R(P)$ under the constraint that $[T^{-1}W_c^0 T^{-T}]_{ii} = 1\ \forall i$.

Using Lagrange multipliers,

$$\begin{aligned} \frac{\partial L}{\partial P} &= \sum_{i=0}^{\infty} \{h_0(i)P^{-1}h_0^T(i) - P^{-1}h_0^T(i)Ph_0(i)P^{-1} + M_o^0 - P^{-1}W_c^0P^{-1} - \lambda P^{-1}W_c^0P^{-1}\} \\ \frac{\partial L}{\partial \lambda} &= \operatorname{tr}(W_c^0 P^{-1}) - n \end{aligned}$$

The solution is via a gradient search method.

$$\begin{aligned} P(k+1) &= P(k) - \mu_1 \frac{\partial L(P,\lambda)}{\partial P}\bigg|_{\substack{P=P(k)\\ \lambda=\lambda(k)}} \\ \lambda(k+1) &= \lambda(k) - \mu_2 \frac{\partial L(P,\lambda)}{\partial \lambda}\bigg|_{\substack{P=P(k)\\ \lambda=\lambda(k)}} \end{aligned}$$

See [62, Chapter 7] for more details.

5.8 Block Processing Filters

The idea for block signal processing dates back to at least 1968 [63]. Consider a standard state-space system. Then, instead of processing a single sample of the input at the time step, block processing suggests processing several samples at once. So the 'blocked' state-space model is

$$x[k+L] = \quad A^L x[k] \quad + \left[A^{L-1}B \,|\, A^{L-2}B \,|\, \cdots \,|\, B \right] \begin{bmatrix} u[k] \\ u[k+1] \\ \vdots \\ u[k+L-1] \end{bmatrix}$$

$$\begin{bmatrix} y[k] \\ y[k+1] \\ \vdots \\ \vdots \\ y[k+L-1] \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ \vdots \\ CA^{L-1} \end{bmatrix} x[k] + \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ CA^{L-2}B & \cdots & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u[k] \\ u[k+1] \\ \vdots \\ u[k+L-1] \end{bmatrix} \tag{5.15}$$

where L is called the blocklength. The actual implementation of block processing requires buffering the input and the output.

One can immediately see where block processing's alternate name – state decimation – comes from. State updates only need to be computed every Lth sample.

The advantages of block processing include:

(i) Reduced multiplications

(ii) Reduced roundoff noise

Zeman and Lindgren [195] showed that the number of multiplications required for each output sample is

$$M = \frac{n(n+L) + L(n + \frac{L+1}{2})}{L}$$

M is minimized for a block length of $L_{opt} = n\sqrt{2}$ where n is the number of state variables. For the optimal block length, the number of multiplications per output sample drops to about $M_{opt} = 3.41n + 0.5$. Thus, even fully parametrized state-space systems only require order n operations per output sample[3], making them much more

[3] Direct Form realizations require the minimum $2n+1$ operations per output sample

cost effective. One can have the roundoff noise performance of a minimum roundoff noise structure with a performance level that's almost as good as the direct form.

Further cost savings in the number of multiplications are made if one blocks a parallel decomposition with each second-order section realized as a minimum noise realization or a normal structure. The optimal block length becomes $L_{opt} = 2\sqrt{n}$, and the number of multiplications per output sample drops to $2n + 2\sqrt{n} + 0.5$.

Rather obvious from the notation, pole sensitivity will also drop, since the eigenvalues of A^L are the eigenvalues of A, each raised to the Lth power. Poles clustered near $z = 1$ will move away from that point. Complex eigenvalues will move both in magnitude and angle, spreading apart and fanning out. If L is increased enough, they will circle around and start returning towards the positive real axis. It would be interesting to characterize the distance between closely spaced eigenvalues as block length changes.

Barnes and Shinnaka [14] give a detailed derivation of the reduction in roundoff noise (per output sample). The noise variance goes down to

$$\sigma^2 \leq \frac{1}{L}\sigma_y^2$$

where σ_y^2 is the output error variance in the SISO case. Thus, output noise drops by a factor of L. The MIMO blocked systems inherit many of the properties of the SISO system that make up the individual blocks. For example, applying a transformation T to (A, B, C, D) results in a blocked system $(T^{-1}A'T, T^{-1}B', C'T, D')$.

Blocking opens up many possibilities for improving performance in a filter.

Parallel computation of the entire output block of L samples requires the same amount of computation time as that for a conventional filter (even with pipelining) to deliver one output sample. One could imagine different scenarios where this could seriously improve performance such as in a time-shared signal processing system.

Another idea suggested in [14] is that a multi-rate blocked structure could be used to utilize longer block lengths for sensitive poles that are close to $z = 1$ (to reduce roundoff noise to an acceptable level) and shorter blocks for poles that are closer to the origin.

Applying blocking to digital compensators raises new, more interesting questions. Surprisingly, I did not come across any work in the FWL literature that investigates this idea. So what happens in the controls setting? The latency for getting a single output goes up to $L * T_s$ where T_s is the sampling rate of the original compensator. Thus, unless T_s is increased by a factor of L, the phase margin of the closed-loop system will significantly decrease due to the added time delay. In signal processing, latency is usually not important. What is important is throughput, and that is maintained constant at one sample per T_s. What if the sampling rate were increased to T_s/L?

If sampling rate is increased, another consideration becomes important: If the time constants of the plant are relatively slow, sampling too fast will result in successive samples being almost equal (and thus, their quantization errors will be highly correlated). The roundoff noise model that is used to predict FWL performance then breaks down. All of these different options require much more thorough investigation and would be fruitful areas of research.

One cautionary note: Coefficient quantization results in the blocked system being (slightly) periodically time-varying. Zeman and Lindgren [195] reported that these effects become neglible for coefficient wordlengths of 16 bits or more. Reng *et al.* [150] also analyze 'aliasing' effects in periodically time-varying blocked systems.

The earlier works on block processing include [26, 27, 115, 116, 118, 181]. See [135, 136] for extensive details on the scattered look-ahead (SLA) and decomposition techniques. Parhi and Messerschmitt [135] describe how to pipeline recursive filters for efficient implementation in VLSI design. They also mention some of the associated finite wordlength effects but state that additional research is required. In [136], a companion paper, they combine scattered look-ahead, clustered look-ahead, decomposition, and incremental output computation to extensively pipeline filters and achieve very high throughput. Parhi in [134] presented a short analysis of FWL effects in SLA filters and concludes that SLA filters have good FWL performance. Chang and Bliss [34] extensively analyzed FWL effects in SLA filters and characterized both roundoff noise and coefficient quantization effects (using the stochastic model). [85,86,106] include additional ideas for pipelining and improving throughput rates for filters. These design techniques all support the statement in the introduction from [147], that for filter designs, good finite wordlength properties are desirable, but the designs must also have other characteristics that make VLSI implementations simpler and cheaper.

Again, it would be interesting to see what kinds of applications such high-throughput filter designs would have in a controls context (if any at all).

5.9 Discrete (or Digital) Redesign

With the topic of discrete redesign, I move one step back in the design hierarchy of Figure 3-4. Discrete redesign's goal is to generate a DT TF from a CT TF. Steiglitz [162] investigated maps from the s-plane to the z-plane. The goal of discretization is to find the mapping that will most closely preserve the CT filter's performance characteristics in the DT filter i.e. to minimize the error in Figure 5-7. As before, the magnitude of the error can be measured in many different ways.

Filter Discretization

Discrete filter redesign is covered extensively in many signal processing and filter design texts, and I refer the reader to them [24,81,130]. Surveying filter discretization techniques could be the topic of another thesis altogether. I will only briefly mention one of the simplest and most common techniques, the bilinear transformation. What is more relevant to my thesis, however, is the relationship of discretization to FWL effects. As I mentioned in the design process chapter, discretization is usually separated from FWL optimization. Thus, a design is discretized with regard to its DT performance, but *without* any consideration of its FWL performance. Minimizing the error in Figure 5-8 would be a more ideal solution than the two-step process of minimizing the error in Figure 5-7 followed by FWL optimization of $K_d(z)$.

Controller Discretization

As mentioned in Chapter 3, a controller is often treated as a filter and optimized for FWL implementation. However, this technique is suboptimal and does not take closed-loop performance into account. Similarly, at the discretization stage, one can treat the compensator as a filter and discretize it for optimal performance, or one can take the entire closed-loop into account and discretize the CT compensator for optimal closed-loop performance.

There are also different levels of sophistication for taking the closed-loop into account. Refer to Section 3.2 for more details.

Exciting new controller discretization techniques, introduced over the last decade, show promise for stimulating further research and better designs. I did not extensively survey the complete literature for filter or controller discretization, but I did search for some of the more recent research in controller discretization, and the following subsections present some of what I found.

5.9.1 Bilinear Transform

The earliest CT filter redesigns used the bilinear transform (also called the Tustin transform or Tustin's method). The transform isomorphically maps the s-plane to the z-plane. The $j\omega$ axis maps to the unit-circle, and the left half-plane maps to the interior of the unit-circle [162]. Thus, a stable CT filter results in a stable DT filter. The transform is defined as

$$H(z^{-1}) = H(s)\Big|_{s\to\frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}}$$

or equivalently,

$$H(z) = H(s)\Big|_{s\to\frac{2}{T}\frac{(z-1)}{(z+1)}}$$

Note that the transform is a function of the sampling frequency. In fact, Kaiser's analysis (see Appendix C.3) showed that as the sampling frequency gets higher, the poles and zeros of the DT TF tend to cluster more tightly, significantly degrading

FWL performance. Tightly clustered poles increase TF sensitivity, thus requiring a longer wordlength to guarantee stability.

The bilinear transform has the property that

$$Z(H_1(s)H_2(s)) = Z(H_1(s))Z(H_2(s))$$

where $Z(\cdot)$ represents the bilinear transform operator [88]. I will now move to discussing controller discretizations. For more discussion about other filter discretizations such as the impulse invariant, step invariant, and ramp invariant methods, refer to the digital signal processing texts mentioned earlier.

5.9.2 Controller Discretization

Tabak [168] applied discretization to a controller using the bilinear transform (treating it as a filter). He noted that the controller's sampling rate usually had to be increased sufficiently high so that the closed-loop performance of the original CT system was maintained. Rattan [148] proposed a redesign that matched the frequency response (at a finite number of points) of the closed-loop system using a CT compensator with that of the closed-loop using a DT compensator (and sample-and-hold hardware).

Several controller redesign techniques compute the DT feedback gain matrix, F_d, using the CT plant and the CT feedback gain matrix, F_c. F_c has presumably already been designed to place the closed-loop poles at desirable locations. The CT system setup is shown in Figure 5-9 and the desired DT setup is shown in Figure 5-10. Kuo [95] and Tsai *et al.* [176] both proposed different methods to compute the feedback and feedforward gain matrices for the digital system, E_d and K_d (see Figure 5-10). Their goals were to match the states at the sampling instants of the system using the CT compensator with those of the system using a DT compensator. Shieh *et al.* [155] proposed a 'locally optimal' redesign method that results in feedback gain matrices which minimize the quadratic cost function

$$J_k \triangleq J(kT) = \frac{1}{2}e(kT+T)^T Qe(kT+T) + \frac{1}{2}\int_{kT}^{kT+T} e(t)Re(t)\ dt$$

where $Q, R \geq 0$ are symmetric weighting matrices. The first part of J_k represents the error at the sampling instant while the second term represents intersample error.

In [154], Shieh *et. al.* describe another digital redesign that they state is based on the "Law of Mean" or the mean value theorem. However, the claim that the redesign is based on is not, in general, true. The generalized version of the mean value theorem states

$$\int_a^b f(x)g(x)\ dx = f(c)\int_a^b g(x)\ dx$$

where $c \in [a,b]$ and with the assumption that $g(x)$ does not change sign over (a,b), $f(x)$ is continuous, and both are integrable on the open interval.

They claim (without proof) the following: "The above relationship can be closely extended to a matrix-valued function case as follows:

$$\int_{kT}^{kT+T} e^{A(kT+T-\tau)}Bu_c(\tau)\ d\tau = u_c(t_\nu)\int_{kT}^{kT+T} e^{A(kT+T-\tau)}B\ d\tau$$

where $0 \leq \nu \leq 1$ and $t_\nu = kT + \nu T \in (kT, kT+T)$"; T is the sampling rate.

This claim is, in general, false for cases where there is more than one state, i.e. in the non-scalar case. The rest of their development does not rely on this generalization, though it is the *motivation* for the work.

They suggest an error criterion based on time response to select the best particular sampling instant within the sampling interval. The cost function is

$$J_e(\nu) = \sum_{i=1}^{n}\left(\int_0^{t_f} |x_{c_i}(t) - x_{d_i}(t)|\ dt\right)$$

where x_{c_i} and x_{d_i} are the ith states of the *plant* with a CT compensator and a DT compensator, respectively; t_f is the finite time of interest over which to match the responses; ν, titled a tuning parameter by the authors, is the percentage of the interval T after which the sampler samples. They demonstrate their scheme with a system tuned with a step input and responding to a step input. It is interesting to ask how the system would respond to an input that it had not been "trained" on?

5.9.3 Fast discretization of the plant

Keller and Anderson [92] describe a novel technique to incorporate a continuous-time plant into a discrete-time controller redesign for a sampled-data system. The idea derives from the signal processing concept of 'blocking' (see Section 5.8). The plant is fast-sampled at a super-multiple of the controller sampling frequency and then 'lifted' to form a MIMO LTI system. Chen and Francis [36] have also intensely explored fast-sampling followed by lifting. Here, I discuss fast discretization as proposed by [92].

What is the advantage of a faster discretization (or equivalently, a better plant approximation)? Normally, the plant is discretized at the controller's sampling frequency. For one thing, sampling faster allows the optimization routine or measurement function to take intersample behavior into account. Thus, a better approximation of the plant allows a sensitivity function to more accurately measure closed-loop performance.

Fast sampling the plant, however, makes the resulting closed-loop system multirate. Blocking is the tool that brings it back into the LTI domain, as an LTI MIMO system.

Consider Figure 5-11. Let the controller sampling rate be T_k and the plant discretization rate $T_p = T_k/L$. The zero-order hold (ZOH) discretization of the plant (A_p, B_p, C_p, D_p) will be

$$\begin{aligned} F_p &= e^{A_p T_p}; \quad G_p = \int_0^{T_p} e^{A_p \tau} B_p \; d\tau; \\ H_p &= C_p; E_p = D_p; \end{aligned}$$

Next, we perform the blocking operation using the blocking equation (5.15), substituting $A = F_p, B = G_p, C = H_p,$ and $D = E_p$.

Now, we must remedy the apparent incompatibility of signal dimensions and rates in the closed-loop: The *Hold – Plant – Sample* combination outputs L samples every T_k seconds while the controller only reads in one sample every T_k seconds. The solution is to insert two devices into the closed-loop. A decimator (commonly used in signal processing) with rate K outputs one sample every K time units and throws away its input at the other $K - 1$ time units. Thus, in our setup, we should insert a *decimator* with rate L between the blocked plant output and the controller input. On the output of the controller, we install a *repeater* which simply outputs the same value every T_k/L seconds. It receives a new value every T_k seconds. Then, the entire decimator-controller-repeater combination could be written as

$$\text{Repeater}\left\{\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} K_d(z_k)\,[1\ 0\ \cdots\ 0]\right\}\text{Decimator}$$

z_k and z_p are simply the transform variables associated with the different delay times, T_k and T_p. In the limit as $L \to \infty$, we recover the CT plant and the standard sampling and ZOH operators from the decimator and repeater respectively.

For multi-variable systems, each 1 in the decimator and repeater would be replaced by an identity matrix, and each 0 by a zero matrix of appropriate size.

Using fast discretization for filter and controller design

Using fast discretization, we can easily reduce the problem of controller design to an H_∞ problem (see Figure 5-12) that can then be solved with standard software. (Actually, it is inaccurate to say that *because* of fast discretization, we can use the H_∞ framework. We could have used the H_∞ framework before, with a CT component discretized at the controller's sampling frequency. The fast discretization will simply produce a much better controller, since the approximation of the plant and other CT components is better, and can account for arbitrarily fast intersample ripples. The difference between the DT controller approximation and the CT controller is fed in as a perturbation, Δ, of the controller. $\Phi(s)$ is an anti-aliasing filter (see Figure 5-13).

Define the cost function

$$\begin{aligned} J_c &\triangleq \Delta(I + P(s)K(s))^{-1}P(s) \\ &= \Delta W(s) \\ \|J_c\| &= \max_{u\in(L_2[0,\infty))} \frac{\|J_c u\|_2}{\|u\|_2} \end{aligned} \qquad (5.16)$$

where $\Delta = (K(s) - H_{T_k}K(s)\Sigma_{T_k}\Phi(s))$. The small-gain theorem requires

$$\|J_c\| < 1 \qquad (5.17)$$

for the loop to be BIBO stable. The controller design objective is to find a $C_d(z_k)$ not only to satisfy (5.17) but to minimize $\|J_c\|$.

Fast discretizing the CT operators $W(s)$, $\Phi(s)$, $H_{T_k}(s)$, and $K(s)$, we obtain the DT operator, J_d. J_d converges to J_c as sampling time tends to 0 (the sampling time for the approximations, *not* for the DT controller $K_d(z_k)$). See [92] for the proof.

Using multi-rate filtering solutions

Let $\tilde{K}(z_k), \tilde{\Phi}(z_k)$, and $\tilde{W}(z_k)$ represent the blocked CT controller, anti-aliasing filter, and $(I + P(s)K(s))^{-1}P(s)$, respectively.

Then, rearranging J_d using the blocked discretization,

$$\tilde{J}_d(z_k) = \left(\tilde{K}(z_k) - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} K_d(z_k) \, [1 \; 0 \cdots 0] \, \tilde{\Phi}(z_k) \right) \tilde{W}(z_k) \tag{5.18}$$

where $K_d(z_k)$ is the DT controller running at sampling rate T_k.

To turn this into an H_∞ problem, we need the important result that norms are preserved under the blocking operation. Consider

$$\sum_l u_d(lT_p)^T u_d(lT_p) = \sum_m \tilde{u}_d(mT_k)^T \tilde{u}_d(mT_k)$$
$$\sum_l e_d(lT_p)^T e_d(lT_p) = \sum_m \tilde{e}_d(mT_p)^T \tilde{e}_d(mT_p)$$

which implies that

$$||u_d||_p = ||\tilde{u_d}||_p$$
$$||e_d||_p = ||\tilde{e_d}||_p$$

Thus, the norm of the signal is preserved under the blocking operation, ensuring that the operator induced norms J_d and $\tilde{J}_d$ are equal also.

One of the powerful results that follows from this problem setup and its solution: "A measure for the impact of the sampling time on controller discretization is the value of $||\tilde{J}_d||_\infty$ for an optimal controller $K_d(z_k)$, and the effect of varying sampling time can be easily examined. If, for an optimal controller, $||\tilde{J}_d||_\infty = 1$, an upper bound for the sampling period T_k based on a sufficient condition for stability is reached" [92].

Refer to [153] for a procedure to solve the H_∞ problem for $K_d(z_k)$.

5.10 Minimum LQG Cost Realization

Take the standard discrete-time (MIMO) model of a plant with measurement and output noise added:

$$\begin{aligned} x_p[k+1] &= A_p x_p[k] + B_p u[k] + D_p w_p[k] \\ z_p[k] &= M_p x_p[k] + v_p[k] \\ y_p[k] &= C_p x_p[k] \end{aligned}$$

where v_p and w_p are assumed to be mutually independent, zero mean, discrete white noise sources with covariance matrices V_p and W_p, and z_p is the measurement vector. This model is slightly more general than the one used by most authors, which includes noise in the output, not in a different measurement vector. Including a measurement vector, z_p, allows one to separate the output from the measurement in the cost function. The cost function J is a quadratic function of the output $y_p[k]$ and the control input $u[k]$:

$$J = \lim_{k\to\infty} E\left\{y_p[k]^T Q_p y_p[k] + u[k]^T R u[k]\right\} + \sum_{i=1}^{n} \frac{\rho_i}{q_i}$$

If the measurement and output are the same, then minimizing the cost function also minimizes the measurement, but this is not a requirement. In fact, in the actual cost function, the difference will be reflected in the weighting matrix that multiplies the measurement noise. If z_p and y_p are the same, then the noise is multiplied by the weighting matrix Q, while if z_p is different from y_p, the measurement noise is weighted by the matrix R.

The desired controller is of the form

$$\begin{aligned} x_c[k+1] &= A_c x_c[k] + B_c z_p[k] \\ u[k] &= C_c x_c[k] + D_c z_p[k] \end{aligned}$$

Now, we introduce the quantization effects. Of the handful of papers that address this problem, most models in the current literature on FWL effects in LQG controllers include roundoff error but ignore coefficient quantization. Liu *et al.* [105] have solved the problem of optimal compensator design for an LQG cost that takes into account roundoff error. Before delving into their solution, it is important to discuss the other works on compensator design with the LQG criterion, as they include ideas that could be used to extend Liu *et al.*'s work.

The first major work to incorporate FWL errors in LQG design was Moroney *et al.* [120]. They solved the following problem: Under infinite-precision optimal control, assume the cost value is J_0. Then, find the minimal wordlength required to implement the controller such that the degradation of the cost function is less than some amount, E_0. They introduce coefficient quantization error into the cost function using the stochastic coefficient quantization model. They develop the statistical wordlength (SWL) to find the minimum wordlength that will, with a certain probability, achieve the cost degradation bound (see Section 4.6 for a discussion of the stochastic coefficient quantization model).

They also present an algorithm to find the transformation to minimize the *increase* in J_0 due to roundoff noise. Note how this differs from Gevers and Li's closed-loop noise gain criterion. Gevers and Li minimize the noise gain of the closed-loop system for roundoff errors while Moroney *et al.* minimize the increase in J due to roundoff noise. It would be interesting to compare how the resulting optimal realizations perform under the two measures (i.e. the performance of the G_{cl}-optimal realization measured through J, and vice versa).

Both these algorithms are similar in that they restrict their search space to that of similarity transformations, i.e. they use the gain matrices computed by using the infinite precision Riccati equations and then find the transformation that will minimize the degradation in the performance measure.

Williamson and Kadiman [186] proposed an iterative *design* scheme:

(i) Compute optimal gains F, L for the plant using modified design equations

(ii) Compute the optimal transformation based on the gains to transform the plant model

(iii) Apply the transformation to the plant model and return to step (i)

Their scheme is much closer to an ideal *design* scheme since it does not restrict the search for optimal gains over the set of similarity transformations on the compensator. However, it does not guarantee an *optimal* solution, either.

Next, I will develop Liu *et al.*'s solution to the LQG FWL implementation design problem. They give necessary conditions for the design to be optimal and then present an algorithm to find the controller matrices that will satisfy these conditions. It is unknown if Williamson and Kadiman's iterative scheme converges to a solution that satisfies these necessary conditions and hence is optimal.

Gevers and Li present a closed-loop sensitivity measure, but this algorithm simply searches for the optimal transformation, not the optimal design. They do suggest that one could formulate the design problem as follows: minimize the H_∞ norm of the difference between the transfer function of the system with an ideal controller and one with an FWL controller. Minimizing this norm over all possible F and L matrices would then give the optimal design. Such a formulation would however ignore roundoff noise.

Liu *et al.* [105] choose to *not* optimize with respect to coefficient quantization errors directly since, they claim, these errors are multiplicative and lead to non-linear equations. However, they do not consider modeling the coefficient quantizations errors stochastically. Such an approach may in fact allow a global solution to minimum cost in the presence of *both* coefficient quantization and roundoff noise. In fact, Gevers and Li [62] state that it should be relatively straightforward to incorporate stochastic coefficient quantization into Liu *et al.*'s design framework.

Returning to the models, taking state and output quantization into consideration changes them to

$$\begin{aligned} x_p[k+1] &= A_p x_p[k] + B_p Q[u[k]] + D_p w_p[k] \\ z_p[k] &= M_p x_p[k] + v_p[k] \\ y_p[k] &= C_p x_p[k] \end{aligned}$$

and

$$\begin{aligned} x_c[k+1] &= A_c Q[x_c[k]] + B_c Q[z_p[k]] \\ u[k] &= C_c Q[x_c[k]] + D_c Q[z_p[k]] \end{aligned}$$

where $Q[\cdot]$ represents the quantization operator. Assume that the matrices $A_c, B_c, C_c,$ and D_c can be implemented exactly. Modeling roundoff error as uniform white noise with zero mean, let

$$\begin{aligned} e_{x_c}[k] &= x_c[k] - Q[x_c[k]] \\ e_{z_p}[k] &= z_p - Q[z_p] \\ e_u[k] &= u[k] - Q[u[k]] \end{aligned}$$

with covariance matrices

$$\begin{aligned} E_{x_{ij}} &\triangleq q_i \delta_{ij} \;\text{ where } q_i = \frac{1}{12} 2^{-2\beta_i} \\ E_z &\triangleq q_z I \;\text{ where } q_z = \frac{1}{12} 2^{-2\beta_z} \\ E_u &\triangleq q_u I \;\text{ where } q_u = \frac{1}{12} 2^{-2\beta_u} \end{aligned}$$

where β_i is the wordlength for the ith state variable and β_z and β_u are the fractional parts of the wordlengths for the A/D and D/A respectively. One of the extensions in Zhu *et al.*'s paper [196] was to assume that each A/D and D/A could be assigned a different wordlength i.e. E_z and E_u would change to

$$\begin{aligned} E_z &\triangleq q_{z_i} \delta_{ij} \;\text{ where } q_{z_i} = \frac{1}{12} 2^{-2\beta_{z_i}} \\ E_u &\triangleq q_{u_i} \delta_{ij} \;\text{ where } q_{u_i} = \frac{1}{12} 2^{-2\beta_{u_i}} \end{aligned}$$

where each of the β_{z_i} and β_{u_i} indicate the fractional parts of the wordlengths of the ith A/D and D/A converter respectively. I shall continue the development without this generalization.

$$\begin{aligned} x_p[k+1] &= A_p x_p[k] + B_p(u[k] - e_u[k]) + D_p w_p[k] \\ z_p[k] &= M_p x_p[k] + v_p[k] \\ y_p[k] &= C_p x_p[k] \end{aligned}$$

and

$$x_c[k+1] = A_c(x_c[k] - e_{x_c}[k]) + B_c(z_p[k] - e_{z_p}[k])$$
$$u[k] = C_c(x_c[k] - e_{x_c}[k]) + D_c(z_p[k] - e_{z_p}[k])$$

The LQG cost function to minimize is

$$J = \lim_{k\to\infty} E\left\{y_p[k]^T Q_p y_p[k] + u[k]^T R u[k]\right\} + \sum_{i=1}^{n} \frac{\rho_i}{q_i}$$

where n is the number of states, Q_p and R are positive definite weighting matrices, ρ_i is the weighting factor for the penalty on the wordlength of the ith state variable, β_i, and q_i is the variance of the error in the ith state variable ($q_i = \frac{1}{12}2^{-2\beta_i}$).

Combining the plant and controller states into one state vector, we get

$$\begin{aligned} x[k] &= \begin{bmatrix} x_p[k] \\ x_c[k] \end{bmatrix}; A = \begin{bmatrix} A_p & 0 \\ 0 & 0 \end{bmatrix}; B = \begin{bmatrix} B_p & 0 \\ 0 & I \end{bmatrix}; \\ y &= \begin{bmatrix} y_p[k] \\ u[k] \end{bmatrix}; C = \begin{bmatrix} C_p & 0 \\ 0 & 0 \end{bmatrix}; D = \begin{bmatrix} D_p \\ 0 \end{bmatrix}; \\ G &= \begin{bmatrix} D_c & C_c \\ B_c & A_c \end{bmatrix}; \; I_0 = \begin{bmatrix} 0 & 0 \\ I & 0 \end{bmatrix}; \; I_1 = \begin{bmatrix} I \\ 0 \end{bmatrix}; I_2 = \begin{bmatrix} 0 \\ I \end{bmatrix}; \\ M &= \begin{bmatrix} M_p & 0 \\ 0 & I \end{bmatrix}; Q = \begin{bmatrix} Q_p & 0 \\ 0 & R \end{bmatrix} \end{aligned} \tag{5.19}$$

with a closed-loop system described as

$$\begin{aligned} x[k+1] &= (A + BGM)x[k] + \begin{bmatrix} D & BGI \end{bmatrix} \begin{bmatrix} w_p[k] \\ v_p[k] \end{bmatrix} + \begin{bmatrix} BGI_2 & BGI_1 & BI_1 \end{bmatrix} \begin{bmatrix} e_x[k] \\ e_z[k] \\ e_u[k] \end{bmatrix} \\ y[k] &= (C + I_0GM)x[k] + I_0GI_1v_p[k] + I_0GI_2e_x[k] + I_0GI_1e_z[k] \end{aligned} \tag{5.20}$$

and the simplified cost function

$$J = \lim_{k\to\infty} E\{y[k]^T Q y[k]\} + \rho^T \beta$$

where ρ is the vector of weights and β is the vector of variances q_i. Substituting the expression for $y[k]$ into J,

$$\begin{aligned} J = {} & \mathrm{tr}\,(X[C + I_0GM]^T Q[C + I_0GM]) \\ & + \mathrm{tr}\,(V_p(I_0GI_1)^T Q(I_0GI_1)) \\ & + \mathrm{tr}\,(E_x(I_0GI_2)^T Q(I_0GI_2)) \\ & + \mathrm{tr}\,(E_z(I_0GI_1)^T Q(I_0GI_1)) + \rho^T\beta \end{aligned}$$

where X is the state covariance matrix satisfying the modified Lyapunov equation

$$\begin{aligned} X = {} & (A + BGM)X(A + BGM)^T + DW_pD^T + (BGI_1)V_p(BGI_1)^T + BI_1E_u(BI_1)^T \\ & + (BGI_1)E_z(BGI_1)^T + (BGI_2)E_x(BGI_2)^T \end{aligned}$$

Since the above equation is linear, we can split it into $X = X_w + X_e$ where X_w represents the error due to disturbances and roundoff errors in the A/D and D/A, and X_e represents the error solely due to roundoff noise. This will allow us to also split the cost term into $J = J_w + J_e$. Note that the cost due to disturbance inputs and A/D and D/A errors will be independent of the coordinates of the controller. Hence, only the J_e portion of the cost will depend on the controller's realization. Including the standard l_2 scaling constraint, the optimization problem can be restated as

$$\min_{G,\beta_i} J = \min_{G,\beta_i} J_w + J_e$$

subject to the constraint

$$[X_w(2,2)]_{ii} = \alpha$$

where α is a scaling factor (α can be thought of as the number of standard deviations that are representable in the register of the ith state variable). See the discussion in Section 2.3 for more details about scaling.

The scaling constraint complicates the design equations. Liu *et al.* break the problem up into two separate ones: finding the optimal controller G and wordlengths β_i and then finding the transformation T such that the scaling constraint is also satisfied. Thus,

$$\begin{aligned}\min_{G,T,\beta_i} J = \min_{G,T,\beta_i}(J_w + J_e) &= \min_{G,\beta_i}\left[\min_T(J_w + J_e)\right] \\ &= \min_{G,\beta_i}\left[J_w + \min_T J_e\right] \qquad (5.21)\end{aligned}$$

The overall algorithm to determine the optimal controller, LQG_{FW}, is:

1) solve for β_i and G using a gradient search and equations given in [105];

2) compute the optimal transformation, T, based on the above G;

3) $\tilde{G} = \begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} G \begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix}$ is the optimal LQG_{FW} controller.

Zhu *et al.* [196] give two additional equations that are included in the gradient search and these equations give the wordlengths for the fractional parts of the A/D and D/A converters. Their paper also allows for skewed sampling.

Refer to Liu *et al.* [105] and Zhu *et al.* [196] for detailed design equations, algorithms, and proofs.

As mentioned before, both these schemes allow one to do *partial* model order reduction. If a particular state is not that important, it will be assigned less bits, instead of being completely eliminated. Meanwhile, important states will be allocated more bits. As Liu *et al.* put it, "[W]e allow the mathematics to assign the appropriate (small or large) computational resources to [each state]."

It would be particularly interesting to extend Liu *et al.*'s LQG framework to include coefficient quantization through the use of a stochastic coefficient quantization error model. Ideally, one would like an algorithm that could output the number of bits that should be allocated to each state variable *and* coefficient.

5.11 Sparseness Considerations

Bomar and Hung [23], extending Chan's work [33], present an algorithm that starts with an initial realization that is optimal (in terms of roundoff noise performance) and then "leads" entries (of the system matrices) closest to powers-of-2 to the power-of-2 while keeping the transfer function constant. The algorithm maintains the l_2 scaling contraint during the procedure, and roundoff noise is kept at an almost optimal level. The resulting state-space realization requires only $(3n + 2)$ multiplies and requires $(n^2 - n - 1)$ powers-of-2 shifts.

The key to Bomar and Hung's algorithm is a transformation matrix $T(t)$ that takes the system to the destination form as it evolves as a function of its parameter, t. In each iteration, a selected non-trivial coefficient is targeted. At each time step, the matrix $T(t)$ is used to transform the system $(A(t), B(t), C(t))$ to $(A(t+1), B(t+1), C(t+1))$ [4]. Then, derivatives of all the relevant matrices (see below) are recomputed, and the next update is then performed. Ultimately, an entry of either $A, B,$ or C reaches a target value, and the algorithm restarts with a new non-trivial coefficient. This process continues for $(n^2 - n - 1)$ steps.

Define the transformation, $T(t)$, as a continuously differentiable function of the variable t. Then, the state-space system can be defined as

$$\begin{aligned} x[k+1] &= A(t)x[k] + B(t)u[k] \\ y[k] &= C(t)x[k] + Du[k] \end{aligned}$$

where

$$\begin{aligned} A(t) &= T(t)^{-1}A(0)T(t) \\ B(t) &= T(t)^{-1}B(0) \\ C(t) &= C(0)T(t) \end{aligned}$$

The main idea is to let the system evolve along the path dictated by its derivatives towards the system with powers-of-2 coefficients that are the valleys. We control the direction of evolution by directing it towards a decreasing derivative that still maintains the l_2 scaling constraint and keeps the observability Gramian (or equivalently, the roundoff noise) about the same.

[4] The evolution actually occurs in continuous time so the indices are a slight abuse of notation.

Using (5.7), the observability and controllability Gramians change as follows:

$$W_c(t) \to T(t)^{-1}W_c(0)T(t)^{-T}; W_o(t) \to T(t)^T W_o(0)T(t)$$

$$W_c(t)W_o(t) \to T(t)^{-1}W_cW_oT(t)$$

$W_c(t)$and $W_o(t)$ can of course also be written as $\sum\limits_{k=0}^{\infty}[A(t)^kB(t)][A(t)^kB(t)]^T$ and $\sum\limits_{k=0}^{\infty}[C(t)^TA(t)^k][C(t)A(t)^k]^T$, respectively.

The scaling conditions remain the standard l_2 constraint: $[W_c(t)]_{ii} = 1$ for $i = 1, 2, ..., n$ (or $[W_c(t)] = \alpha$).

The t-varying variance is

$$\sigma_y^2(t) = \nu[1 + \mathrm{tr}\,(W_o(t))]\sigma_e^2$$

where $\nu = 1$ for roundoff after summation (and multiplication) and $\nu = n + 1$ for rounding before addition.

Also, the minimum value of $G(t)$ is given by

$$G(t) = \frac{1}{n}\left(\sum_{i=1}^{n}\nu_i(t)\right)^2$$

where the $\nu_i(t)$ are the eigenvalues of the product $W_c(t)W_o(t)$. Analogous to the case of constant system matrices, the necessary and sufficient condition for minimum roundoff noise gain is $W_c(t) = \gamma(t)^2W_o(t)$ where

$$\gamma(t) = \frac{n}{\sum\limits_{i=1}^{n}\nu_i(t)}$$

Now, they present the definitions of the derivatives of each of the above terms

$$\dot{A}(t) = -T^{-1}(t)\dot{T}(t)T(t)^{-1}(t)A(0)T(t) + T(t)^{-1}A(0)\dot{T}(t)$$

$$= -T^{-1}(t)\dot{T}A(t) + A(t)T^{-1}(t)\dot{T}(t)$$

$$= A(t)F(t) - F(t)A(t)$$

where

$$F(t) = T^{-1}(t)\dot{T}(t)$$

Similarly,

$$\dot{B}(t) = -F(t)B(t)$$

$$\dot{C}(t) = C(t)F(t)$$

$$\dot{W}_c(t) = -[F(t)W_c(t) + W_c(t)F^T(t)]$$

$$\dot{G}(t) = \operatorname{tr}[F^T(t)W_o(t) + W_o(t)F(t)]$$

The individual elements of $a_{ij}(t)$ and $f_{ij}(t)$ of $A(t)$ and $F(t)$ respectively are

$$\begin{aligned}\dot{a}_{ij}(t) &= \sum_{k=1}^{n} a_{ik}(t)f_{kj}(t) - \sum_{k=1}^{n} f_{ik}(t)a_{kj}(t)\\ &= \sum_{l=1}^{n}\delta_{jl}\sum_{k=1}^{n} a_{ik}(t)f_{kl}(t) - \sum_{l=1}^{n}\delta_{il}\sum_{k=1}^{n} a_{kj}(t)f_{lk}(t)\end{aligned}$$

Switching the order of summation,

$$\begin{aligned}\dot{a}_{ij}(t) &= \sum_{l=1}^{n}\sum_{k=1}^{n}\delta_{jl}a_{ik}(t)f_{kl}(t) - \sum_{k=1}^{n}\sum_{l=1}^{n}\delta_{il}a_{kj}(t)f_{lk}(t)\\ &= \sum_{l=1}^{n}\sum_{k=1}^{n}[\delta_{jl}a_{ik}(t) - \delta_{ik}a_{lj}]f_{kl}(t)\\ &= \sum_{l=1}^{n}\sum_{k=1}^{n} S^A_{lk}(t,i,j)f_{kl}(t)\end{aligned}$$

where

$$S^A_{lk}(t,i,j) \triangleq \delta_{jl}a_{ik}(t) - \delta_{ik}a_{lj}$$

Proceeding similarly for B, C, W_c, and G,

$$\dot{B}_i(t) = \sum_{l=1}^{n}\sum_{k=1}^{n} S^B_{lk}(t,i)f_{kl}(t)$$

$$\dot{C}_i(t) = \sum_{l=1}^{n}\sum_{k=1}^{n} S^C_{lk}(t,i)f_{kl}(t)$$

$$[\dot{W}_c(t)]_{ii}(t) = \sum_{l=1}^{n}\sum_{k=1}^{n} S^{W_c}_{lk}(t,i)f_{kl}(t)$$

$$\dot{G}(t) = \sum_{l=1}^{n}\sum_{k=1}^{n} S^G_{lk}(t)f_{kl}(t)$$

where

$$
\begin{aligned}
S^B_{lk}(t,i) &= -\delta_{ik} B_l(t) \\
S^C_{lk}(t,i) &= -\delta_{ik} C_l(t) \\
S^{W_c}_{lk}(t,i) &= -2\delta_{ik}[W_c(t)]_{lk}(t) \\
S^G_{lk}(t) &= 2[W_o(t)]_{lk}
\end{aligned}
$$

Using the vec operator, the notation simplifies to

$$
\begin{aligned}
\text{vec } \dot{A}(t) &= S^A(t)\text{vec } F(t) \\
\text{vec } \dot{B}(t) &= S^B(t)\text{vec } F(t) \\
\text{vec } \dot{C}(t) &= S^C(t)\text{vec } F(t) \\
[\dot{W}_c(t)]_{ii} &= S^{W_c}(t,i)\text{vec } F(t), \; i = 1,2,...,n \\
\dot{G}(t) &= S^G(t)\text{vec } F(t) \qquad (5.22)
\end{aligned}
$$

where $S^A(t) \in \mathbb{R}^{n^2 \times n^2}$, $S^B(t)$ and $S^C(t) \in \mathbb{R}^{n \times n^2}$, $S^{W_c}(t,i) \in \mathbb{R}^{1 \times n^2}$ and $S^G(t) \in \mathbb{R}^{1 \times n^2}$ are made up respectively of S^A_{lk}, S^B_{lk}, $S^C_{lk}, S^{W_c}_{lk}(t,i)$, and $S^G_{lk}(t)$ as defined in (5.22).

Now, to simplify notation further, introduce the vector of coeffecients of the system matrices:

$$
\Theta(t) \triangleq \begin{bmatrix} \text{vec } A(t) \\ \text{vec } B(t) \\ \text{vec } C(t) \end{bmatrix}
$$

Then, the derivatives can be combined into

$$
\dot{\Theta}(t) = S^\Theta(t)\text{vec } F(t)
$$

where

$$
S^\Theta(t) = \begin{bmatrix} \text{vec } S^A(t) \\ \text{vec } S^B(t) \\ \text{vec } S^C(t) \end{bmatrix}
$$

Finally, introduce the $(m+n+1) \times n^2$ matrix

$$L_m(t) = \begin{bmatrix} S^{W_c}(t,1) \\ \vdots \\ S^{W_c}(t,n) \\ S^G(t) \\ S_1^{\Theta}(t) \\ \vdots \\ S_m^{\Theta}(t) \end{bmatrix}$$

Then, impose the three following constraints

$$[W_c(t)]_{ii} = 1, \; i = 1, 2, ..., n$$

$$G(t) = \mathrm{tr}\,(W_o(t))$$

$$\theta_i(t) = \theta_i$$

which translate to maintaining the original scaling, the original roundoff noise gain, and keeping constant the parameters that have already been set. Thus, the derivatives of each of these terms should be maintained at zero, which results in

$$L_m(t)\mathrm{vec}\ F(t) = 0$$

Thus, vec $F(t)$ must belong to the null space of $L_m(t)$ i.e. vec $F(t) \in \mathcal{N}[L_m(t)]$.

Each time an additional parameter reaches its target, L_m is augmented and $\mathcal{N}[L_m(t)]$ shrinks by one, allowing the process to be repeated at most $n^2 - n - 1$ times.

Note that if the integration step is set too small, it can prevent most coefficients from reaching their destination power-of-two value. This probably means there is no *minimium noise structure* with that many powers-of-two coefficients, and that one must deviate a little from minimum noise in order to reach more powers-of-two coefficients. They go on to describe integration techniques and the actual implementation details in their paper.

Amit and Shaked [3] extend Bomar and Hung's algorithm, setting the target values to be 0 and ±1 instead of powers-of-2. They show that if one uses the *actual* roundoff noise ($\operatorname{tr}((QW_o)+I)$) as the measure to minimize instead of the theoretical roundoff noise ($\operatorname{tr}(W_o)$), the new structures generated by their algorithm are not only sparse but also perform better than the optimal, minimum noise realization structures of Mullis, Roberts, and Hwang when measuring actual roundoff noise.

They also extend the work of Moroney *et. al.* [119, 120], solving the LQG cost minimization problem while including the actual roundoff noise instead of the theoretical noise. It would be interesting to try and incorporate Amit and Shaked's work with that of Liu *et al.* mentioned in Section 5.10. Optimizing for actual roundoff noise would result in a sparse and optimal LQG structure.

Gevers and Li [62] also point out a useful extension to Bomar and Hung's algorithm: By changing $G(t)$ and $\dot{G}(t)$, one can use the same algorithm to find sparse structures that minimize measures other than the actual roundoff noise measure (while maintaining the l_2 scaling constraint).

Smith and Bomar [158] present an algorithm that starts with the Direct Form and then directs matrix entries in order to lower the overall roundoff noise and maintain a scaling constraint. Their method however only results in an upper or lower triangular matrix, so it still requires order n^2 operations per output sample. They suggest that their algorithm is particularly useful for two-dimensional filters, where decomposition into subfilters is not possible.

5.12 Other Optimizations

Many of the optimizations for the corresponding measures in Chapter 4 did not make it into this chapter due to lack of time. I am including citations to the relevant references, and leave the development of these optimizations for future work.

The optimal transformations for the Closed-Loop Noise Gain (G_{cl}), Closed-Loop Sensitivity ($M_{cl,L_{12}}$), and the Closed-Loop Synthetic Measure ($G_{T,cl}$) all appear in [62]. The minimal transformation for the M_2 measure is in [111]. The Closed-Loop Pole Sensitivity minimizing transformation is given in [100].

One of the discrete-time redesign techniques that I did not get a chance to include was that of Fujimoto and Kawamura [57,58]. They use a generalized-hold device for the controller output. They call the overall system N-Delay control.

Generalized hold devices are essentially multi-rate hold devices, i.e. they change their hold output every T/L seconds, where T is the output rate of the controller. However, they are *not* the same as a zero-order hold running at a rate T/L seconds. The generalized hold computes its output for the $(L-1)$ sampling times without any input from the controller. Thus, it outputs a fixed-shape signal but changes the amplitude of each subsample based on a table or a similar device that it contains. For example, let $H_g(T)$ denote the output of a generalized hold with sampling rate T, with $L = 2$ for simplicity, and let $y[k]$ be the input to the hold at time $t = kT$. Then a possible equation to describe $H_g(T)$ may be

$$H_g(T) = \begin{cases} .05y[k] & kT \leq t < kT + T/2 \\ y[k] & kT + T/2 \leq t < kT + T \end{cases}$$

There seems to be no research investigating the impact of fixed-point hardware, or even finite-precision hardware, for generalized holds and their ability to help control systems.

Another area I did not get a chance to include material about is Error Spectrum Shaping (ESS). The idea involves feeding back the state quantization error and has been alternatively called residue feedback. The δ-operator realizations, popularized by Middleton and Goodwin in the 80's [117], also fall into the category of ESS. See [10, 53, 70, 71, 96, 124, 185, 187, 188] for various discussions on FWL effects and ESS.

The next chapter describes these and many other open questions and also suggests a design idea for a software tool to facilitate design.

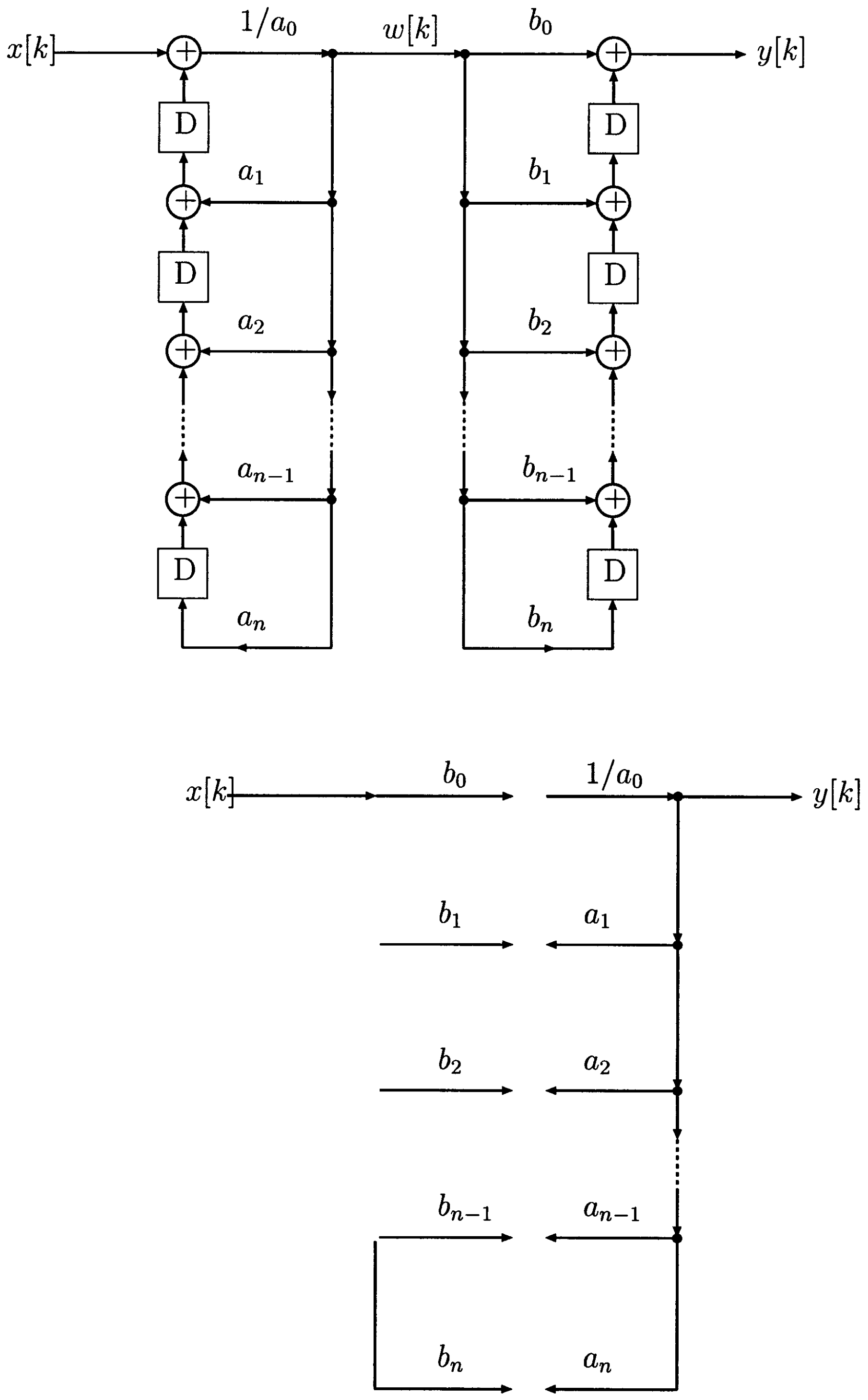


Figure 5-4: Direct Form I Transpose.

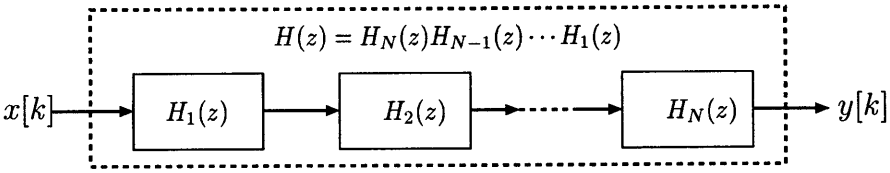


Figure 5-5: Cascade decomposition of $H(z)$.

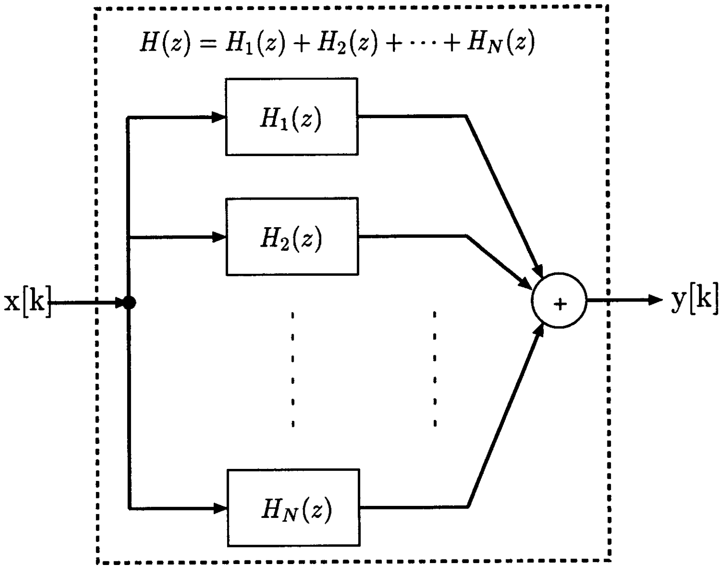


Figure 5-6: Parallel decomposition of $H(z)$.

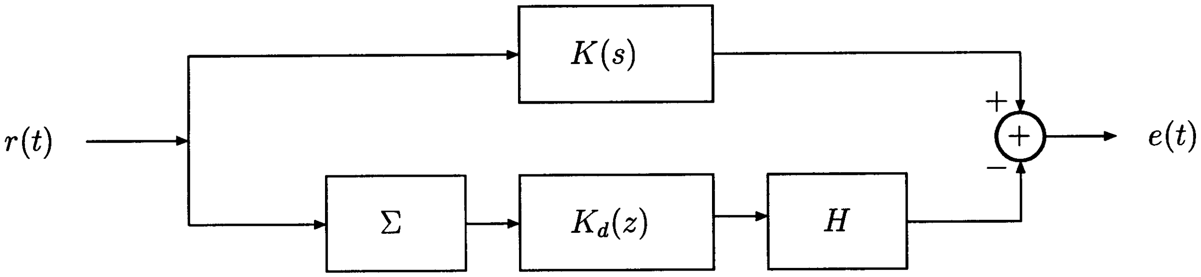


Figure 5-7: FWL filter error.

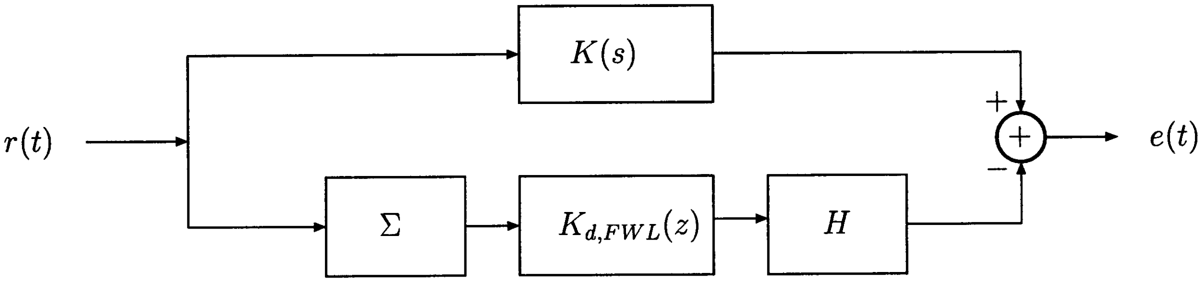


Figure 5-8: FWL DT filter error.

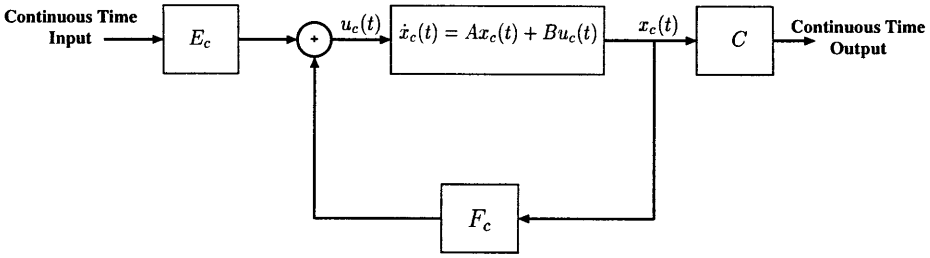


Figure 5-9: Continuous time system.

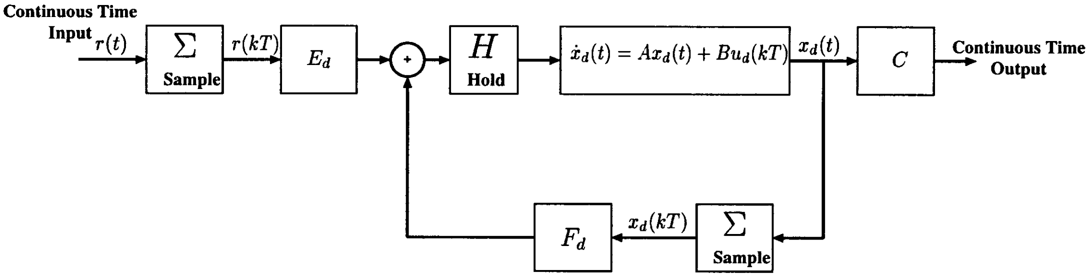


Figure 5-10: Digital redesign.

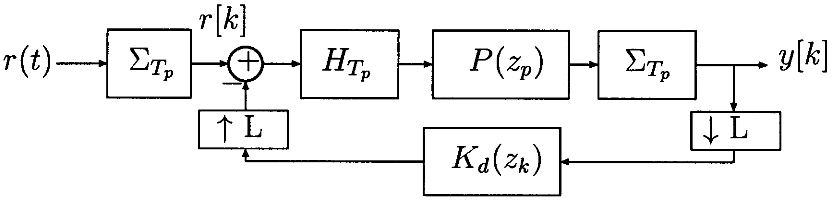


Figure 5-11: A lifted closed-loop system.

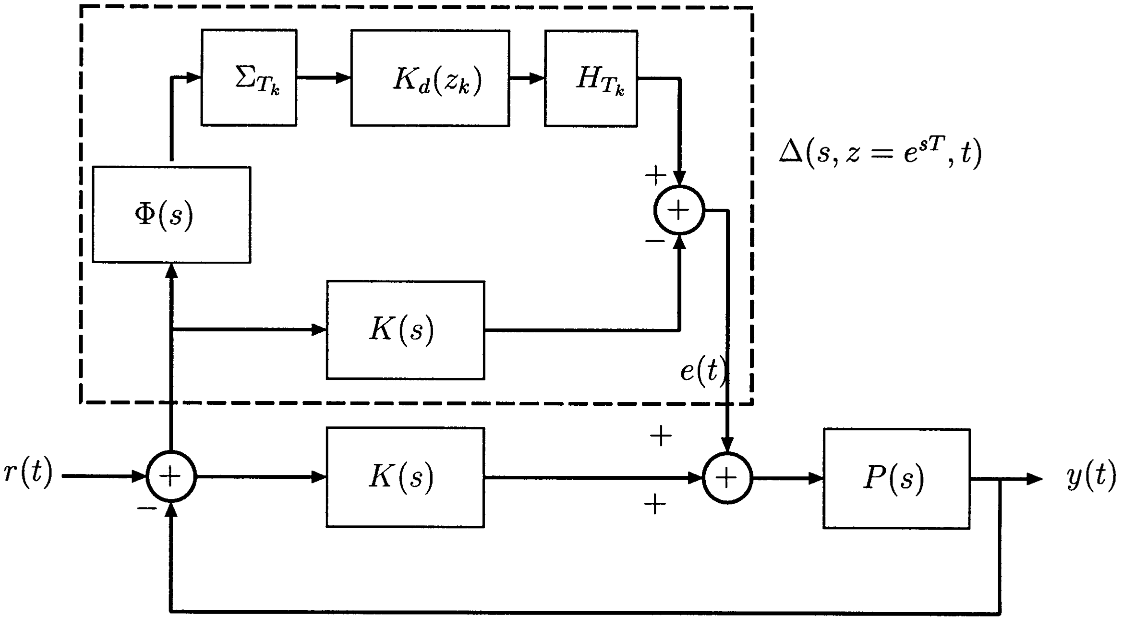


Figure 5-12: DT controller approximation error.

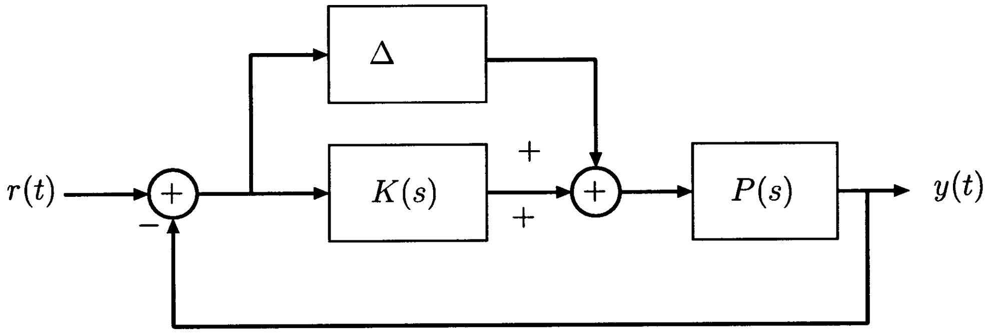


Figure 5-13: Approximation error as a disturbance.

Chapter 6

Conclusion

The woods are lovely, dark and deep,
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep.
—Robert Frost

And so, we near the end of our journey. But, it is far from any end. Research on FWL effects is relatively young, with many new results, especially in the controls context, being published in the last decade. I have hoped to show throughout this thesis the many smaller and larger questions that still loom. Apart from answering these questions and developing the research on FWL effects, a very important direction of development is software.

Currently, there seems to be little available software to explore the many possible design options. Tools like SIMULINK from The MathWorks, Inc. and System Build from ISI help one to evaluate specific designs but do little to facilitate the search of the very large, non-linear design space. What features would the ideal tool have? What kind of paradigm might it operate in? These are perhaps the most critical questions whose answers will move much of the current research onto the desktops of more engineers. I shall speculate on these questions and suggest a possible direction for further work.

Ideally, then, a filter design tool would ask the designer for specifications including memory limitations, speed requirements, accuracy requirements, processor limitations (operations/sec), and of course, the desired frequency and time response characteristics. It would then try out many of the optimizations listed in the previous chapter, partly in a structured way based on the relationships of the optimizations and partly just with a random search. As it tries different optimizations, the tool would also use different measures and apply it to each design. Finally when it has found several possible candidates, it would return them to the designer who could then choose to either use one of them, rerun the search with different parameters, or focus the search to a specific region of the design space using only two or three measures.

Controller design would additionally require plant parameters and design constraints for the sample and hold devices, such as sampling rate, sampling resolution, and the types of possible output holds.

As an abstract description, this tool sounds ideal, the magic wand of fixed-point design. How realistic is it? Some of its lowest level building blocks would be functions to evaluate a given realization with each measure. The functions can be implemented using the algorithms in many of the references in this docuemnt. Next, one would need to implement a heuristic to decide how to search the design space. This open-ended problem's solution could range from random search to an extremely sophisticated non-linear search algorithm. Finally, such a tool would have to have a user interface that would allow easy specification of design constraints, not a simple task either. But the utility of such a tool cannot be doubted. In fact, even a simpler, less feature-rich version of such software would be incredibly useful, in applications engineering and research.

I would like to pose and answer another question: What exactly is this 'design space' that I mention? I would best characterize it by its 'design points'. One point might be an nth order pipelined filter with m zeros, realized with some cascade sections, some sections as lattices, and perhaps another section as a minimum roundoff noise structure. Another point may be an $(n + 5)$th order filter with two sections realized as minimium M_{L_2} sensitivity sections and some sections realized in second-order direct form. One can see the immense number of possibilities. My statement in the introduction, that the available tools *and* computational resources shape our approaches and views of a problem, is reflected in my vision of this software tool.

What are some of the interesting theoretical or mathematical questions? One direction is the modern controls paradigm. Fialho and Giorgiu [55] and Li [100] explored stability robustness and performance. Li's model assumed a DT plant model as did the one in [55]. Along with Keller and Anderson's work [92] and Chen and Francis' explorations [36] of fast sampling and lifting, one should be able to better include the effects of the plant in the closed loop, in turn leading to more accurate measures and better FWL controller designs. The stochastic coefficient quantization model also seems to offer hope for a cohesive roundoff noise/coefficient quantization model. Combining the stochastic models with the modern controls paradigm may also lead one closer to a direct route from CT Plant $\rightarrow$ FWL DT controller. I also mentioned several different possibilities for incorporating block processing in controller design in Section 5.8. Another area that seems completely unexplored is the use of generalized holds with fixed-point hardware. Indeed, the possibilities seem endless.

And these are all ideas that synthesize current research. New research in sampled-data systems and robust controls will surely provide many more directions to explore.

What about for digital filters? One question that I would like to see investigated is that of *coefficient* wordlength: how much can one shorten the wordlength of each coefficient and still maintain a certain performance level? Sung and Kum's [166] optimization routines do exactly this optimization using a brute force search. To put the optimization in an analytical framework, perhaps using the stochastic coefficient quantization error model, should be relatively straightforward (assign each associated random variable a different variance) and may yield interesting results. However, there is some question as to how far one can push this stochastic model [80]. Another approach to filter design may be to apply some of the machinery of robust or H_2/H_∞ controller design, perhaps to yield a direct path from CT filter $\rightarrow$ FWL DT filter. For filters that operate with analog signals, analyzing some of the different sample and hold hardware in the context of FWL effects would be useful. Perhaps one of the most exciting research area is VLSI design and implementation of filters. Unfortunately, I did not have much time to evaluate this area more thoroughly, and I leave it for future work.

Looking towards the future, I bring this journey towards its inevitable end and wish you well, my reader, on your own explorations of the world of finite wordlength effects.

THE END

Appendix A

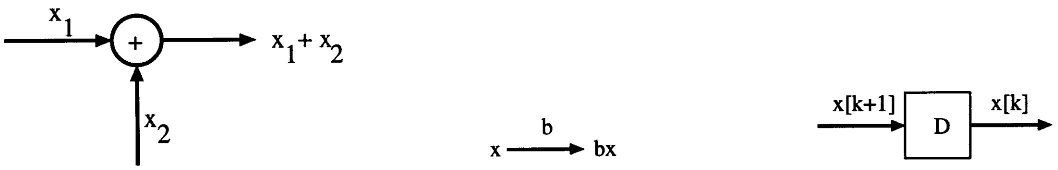
Notation

CT	- Continuous Time
DT	- Discrete Time
DSP	- Digital Signal Processing
	- Digital Signal Processor
FWL	- Finite Word Length
TF	- Transfer Function
$\mathbb{R}^n$	- the space of n-dimensional vectors (the standard Euclidean space).
$\mathbb{R}^{m \times n}$	- the space of real $m \times n$ matrices.
$\mathbb{Z}^n, \mathbb{Z}^{m \times n}$	- the space of n-dimensional vectors and $m \times n$ matrices with integer valued entries.
^{T}	- the matrix transpose operator.
^{H}	- the matrix Hermitian operator (conjugate transpose).
$E[\cdot]$	- Expected value operator.
$[A]_{ij} \equiv a_{ij}$	- ijth entry of the matrix A.
$Q[\cdot]$	- Quantization operator. $Q[x]$ returns the quantized value of x.

Appendix B

Definitions

Arithmetic Component A digital multiplier, adder, or delay element. Depicted notationally as:



(a) Add Element

(b) Multiply Element

(c) Delay Element

Figure B-1: Arithmetic Components

Arithmetic-Geometric Mean Inequality

$$\frac{1}{n}\sum_{i=1}^{n} r_i \geq \left[\prod_{i=1}^{n} r_i\right]^{1/n}$$

where $r_i \in \mathbb{R}$ and $r_i \geq 0,\ i = 1, 2, ..., n$. Equality holds if and only if $r_1 = r_2 = \cdots = r_n$.

Blocking See Section 5.8.

Cauchy-Schwarz Inequality A special case of Hölder's inequality with $p = q = 2$:

$$\sum_{j=0}^{n-1} |b_j c_j| \leq \left(\sum_{j=0}^{n-1} |b_j|^2\right)^{\frac{1}{2}} \left(\sum_{j=0}^{n-1} |c_j|^2\right)^{\frac{1}{2}}$$

where $b_j, c_j \in \mathbb{R}$.

Controllability (or Reachability) Gramian (Reachable means that any state is reachable *from* the origin while controllable means that any state is controllable *to* the origin. The difference between the two is only relevant in the DT context [41].)

Discrete Time:

The positive definite solution W_c to the Lyaponov equation

$$W_c = AW_cA^T + BB^T \tag{B.1}$$

which is

$$W_c = \sum_{k=0}^{\infty} A^k BB^T (A^T)^k \tag{B.2}$$

where the reachable eigenvalues of A all have magnitude less than 1.

Continuous Time:

The positive definite solution W_c to the Lyaponov equation

$$AW_c + W_cA^T = -BB^T \tag{B.3}$$

which is

$$W_c = \int_0^{\infty} e^{\tau A} BB^T (e^{\tau(A)})^T \; d\tau \tag{B.4}$$

where the reachable eigenvalues of A all have negative real parts.

Many authors use the letter K instead of W_c.

Eigenvectors A left eigenvector of a matrix A is a nonzero vector y such that $A^T y = \lambda y$ where λ is the associated eigenvalue. A right eigenvector of a matrix A is a nonzero vector x such that $Ax = \lambda x$ where λ is the associated eigenvalue. A is normal ($A^T A = AA^T$) if and only if X, the matrix of right eigenvectors is orthogonal.

Frequency Transformation A frequency transformation $F(z)$ is a function such that, given a DT filter $H(z)$, the composition $G(z) = H(F(z))$ results in a filter with desired frequency response characteristics. Typically, an acceptable low-pass filter $H(z)$ is given and the problem is to design an $F(z)$ so that the composition has the required pass bands [151, page 203], [24, Table 3.5].

$F(z)$ must have the following properties

1. $F(z)$ should map the unit circle into itself, i.e.

$$F(e^{j\phi}) = e^{j\theta(\phi)}$$

so that the frequency response of $G(z)$ should be related to that of $H(z)$ by the relationship

$$G(e^{j\phi}) = H(e^{j\theta(\phi)})$$

2. If $H(z)$ is stable and minimum phase, then $G(z)$ should have the same properties. If λ is a pole (zero) of G, then $F(\lambda)$ is a pole (zero) of H. Therefore if $|\lambda| < 1$ implies $|F(\lambda)| < 1$, then these properties will be preserved.

Thus, $F(z)$ is a frequency transformation if

$$|z| > 1 \Leftrightarrow |F(z)| > 1$$

$$|z| = 1 \Leftrightarrow |F(z)| = 1$$

$$|z| < 1 \Leftrightarrow |F(z)| < 1$$

Generalized Sample-and-Hold A hold circuit produces a CT output signal $u(t)$ based on the DT input $v[k]$ at $t = kT$. A general hold circuit can be expressed as

$$u(t) = \sum_{k=0}^{\infty} h(t - kT)v[k]$$

where the response function $h(t)$ may be a matrix (in the MIMO case) and T is the hold sampling period.

If $h(t) \neq 0$ only for $t \in (0, T_0)$, the circuit is called a finite-response hold. If $T_0 = NT$, it is called an N-interval hold. The 1-interval hold with a general response function $h(t)$ is called Chammas-Leondes' generalized hold or simply generalized hold [6].

Gramians See *Observability Gramian* and *Controllability Gramian.*

Haddamard's inequality Let A be an arbitrary $n \times n$ nonsingular matrix with real elements. Then

$$(\det A)^2 \leq \prod_{i=1}^{n} \left(\sum_{k=1}^{n} a_{ik}^2 \right)$$

If A is a positive definite, symmetric matrix, we can reduce the above to

$$0 \leq \left[\frac{\det A}{\prod_{i=1}^{n} [A]_{ii}} \right]^{1/2} \leq 1$$

Hankel Singular Values The Hankel Singular Values, ν_i, are the square roots of the eigenvalues of the product of the observability and controllability Gramians:

$$\{\nu_i\} = \sqrt{\lambda_i(W_c W_o)}, \ i = 1, ..., n$$

Note that the Hankel Singular values are invariant under similarity transformations and under frequency transformations [121].

Hessenberg form A matrix $A = \{a_{ij}\} \in \mathbb{R}^{n \times n}$ is in upper Hessenberg form if $a_{ij} = 0$ for $i - j \geq 2$, $i, j \in \{1, 2, ..., n\}$. A 3×3 matrix in Hessenberg form would look like

$$\begin{bmatrix} * & * & * \\ * & * & * \\ 0 & * & * \end{bmatrix}$$

For any matrix A, there exists an orthogonal $T \in \mathbb{R}^{n \times n}$ such that $T^{-1}AT$ is in Hessenberg form [62].

A SISO system realization (A, B, C, D) is in system Hessenberg form if A is of Hessenberg form and all the entries of either B or C are zero except the first one. Every SISO system can be reduced to system Hessenberg form [109]. The proof for Lutz and Hakimi's idea is in [62, page 189].

Hölder's inequality

$$\sum_{i=1}^{u}\prod_{j=1}^{v} a_{i,j}^{r_j} \le \prod_{j=1}^{v}\left(\sum_{i=1}^{u} a_{i,j}\right)^{r_j}$$

where $a_{i,j}$ are nonnegative real numbers for $i = 1, ..., u$ and $j = 1, ..., v$, and the r_j are nonnegative real numbers such that $\sum_{j=1}^{v} r_j = 1$.

This inequality holds as $r_i \to 0$, or equivalently, for $\frac{1}{r_i} \to \infty$.

Renaming the terms and specializing the inequality, it can be reduced to

$$\sum_{j=0}^{n-1} |b_j c_j| \le \left(\sum_{j=0}^{n-1} |b_j|^p\right)^{\frac{1}{p}} \left(\sum_{j=0}^{n-1} |c_j|^q\right)^{\frac{1}{q}}$$

where b_j, c_j are real numbers and $\frac{1}{p} + \frac{1}{p} = 1$.

Hybrid Systems A hybrid system refers to a mixed continuous- and discrete-time system in a single feedback loop. A typical setup is depicted in Figure B-2. Much research in recent years has focused on hybrid systems as use of discrete time controllers has surged. The hold block typically represents a zero-order-hold but can also be a generalized hold function.

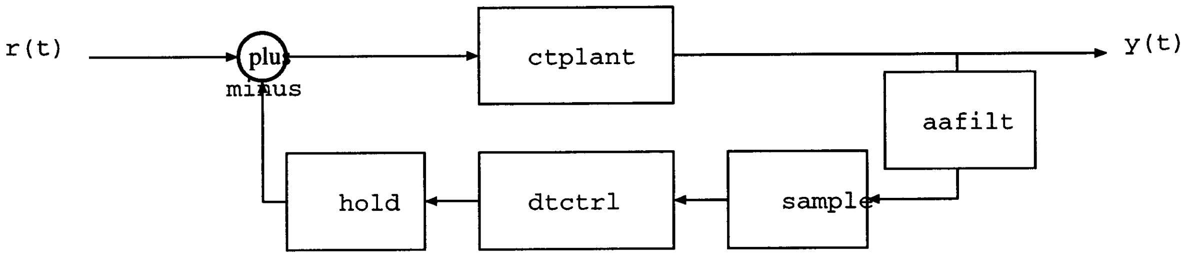


Figure B-2: A Hybrid System

Kronecker Product The Kronecker product of two matrices, A and B, where A is $p \times q$ and B is $m \times n$, is denoted $A \otimes B$:

$$A \otimes B = \left[\begin{array}{c|c|c} a_{11}B & \cdots & a_{1q}B \\ \hline \vdots & \cdots & \vdots \\ \hline a_{p1}B & \cdots & a_{pq}B \end{array}\right]$$

The elements of $A \otimes B$ consist of all possible products $a_{ij}b_{rs}$ and the matrix has dimension $pm \times qn$. See [97].

The LQG Problem The (steady-state) linear quadratic Gaussian (LQG) problem in CT is defined as follows.

Assume the following system description:

$$\dot{x}(t) = Ax(t) + Bu(t) + w_u(t)$$

$$y(t) = Cx(t) + Du(t) + w_y(t)$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n},$ and $D \in \mathbb{R}^{p\times m}$; $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector, and $y \in \mathbb{R}^p$ is the output vector; $w_u \in \mathbb{R}^m$ is the uncorrelated white noise input disturbance with covariance matrix Ξ_u while $w_y \in \mathbb{R}^p$ is the uncorrelated white noise output disturbance with covariance matrix Ξ_y.

Represent the performance of the system with the following scalar quadratic function of the states and the input:

$$J_c = E\left[\lim_{\tau\to\infty} \frac{1}{2\tau}\int_{-\tau}^{\tau} (x^T(t)Qx(t) + u^T(t)Ru(t))\ dt\right] \tag{B.5}$$

where the weighting matrices satisfy $R > 0$ and $Q \geq 0$.

The objective then is to minimize J_c with a compensator.

Latency The total time difference between the time a sample is input into a filter and the time of its corresponding effect on the output.

Lifting See Section 5.9.3

Norms The Frobenius norm of a matrix A is defined as

$$||A||_F^2 = \text{tr}\,(A^T A)$$

and is simply the root sum of the squares of each of the entries of A, i.e. $||A||_F^2 = \sum_{i,j} a_{ij}^2$. Note that the Frobenius norm of a matrix can also be written using the vec operator.

$$||A||_F^2 = (\text{vec } A)^H (\text{vec } A)$$

Signal Norms

The Frobenius norm of $f(e^{j\omega})$ is

$$\begin{aligned}\|f(e^{j\omega})\|_F &\triangleq \left(\sum_{i=1}^{n}\sum_{k=1}^{m}|f_{ik}(e^{j\omega})|^2\right)^{1/2}\\ &= \{\mathrm{tr}\,[f^T(e^{-j\omega})f(e^{j\omega})]\}^{1/2}\end{aligned}$$

The L_p norm of $f(t) \in \mathbb{C}^{n\times m}$ is

$$\|f\|_p = \left(\frac{1}{2\pi}\int_0^{2\pi}\|f(e^{j\omega})\|_F^p\, d\omega\right)^{1/p} \tag{B.6}$$

System Norms

Let $G(s)$ be a $p \times m$ transfer matrix. Then,

$$\begin{aligned}\|G\|_2 &= \left\{\frac{1}{2\pi}\int_{-\infty}^{\infty}\mathrm{tr}\,[G(j\omega)^H G(j\omega)]\, d\omega\right\}^{1/2}\\ &= \left\{\frac{1}{2\pi}\int_{-\infty}^{\infty}\|G(j\omega)\|_F\, d\omega\right\}^{1/2}\\ \|G\|_\infty &= \sup_\omega \sigma_{\max}(G(j\omega))\end{aligned}$$

where $\sigma_{\max}G(j\omega)$ denotes the maximum singular value of $G(j\omega)$. See [45] for more details.

Observability Gramian Discrete Time:

The solution W_o to the Lyaponov equation

$$W_o = A^T W_o A + C^T C \tag{B.7}$$

The solution is

$$W_o = \sum_{k=0}^{\infty}(A^T)^k C^T C A^k \tag{B.8}$$

where the observable eigenvalues of A all have magnitude less than 1.

Continuous Time:

The solution W_o to the Lyaponov equation

$$W_oA + A^TW_o = -C^TC \tag{B.9}$$

The solution is

$$W_o = \int_0^\infty (e^{\tau A})^T C^T C e^{\tau A} \; d\tau \tag{B.10}$$

where the observable eigenvalues of A all have negative real parts.

Many authors use the letter W instead of W_o.

Realization A realization is a particular implementation of a system. It specifies the coefficients and, in many cases, the order of operations.

When using infinite precision, performing a non-singular similarity transformation on a state-space system results in an identical system (in terms of the input/output description). However, its realization has changed since the coefficients have changed. A different order of operations but with the same coefficients, such as the Direct Forms I and II, are also considered different realizations.

Two important issues related to realizations arise in the context of FWL effects:

Non-singular similarity transformations do not change the system to equivalent forms. Some transformations are "better" than others. Searching for the best transformation is in fact a key challenge in finding the "optimal" realization of a system.

With an infinite-precision representation, any non-singular transformation of the state variables results in an equivalent system.

Order of operations is not irrelevant. The exact order of additions and multiplications becomes important since quantization takes place after each operation. Several authors have presented different notations to capture the circuit topology in the matrix notation of the state-space description (e.g. [19,119,151]) and discuss transformations based on the new notation.

Sampled-data System See *Hybrid System.*

Schur form A realization (A, B, C, D) is a Schur realization if the A matrix is in the real Schur form

$$A = \begin{bmatrix} A_{11} & x & \cdots & x & \cdots & x \\ 0 & A_{22} & \cdots & x & \cdots & x \\ \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & A_{ii} & \vdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & A_{mm} \end{bmatrix}$$

where each A_{ii} is either a real number or a 2×2 matrix with complex conjugate eigenvalues [62].

Any matrix $A \in \mathbb{R}^{n \times n}$ can be transformed to a Schur realization by orthogonal similarity transformations. Furthermore, any realization can be transformed to the following optimal Schur realization with at least $\frac{1}{2}n(n-1)$ zero elements. The optimal Schur realization has an A matrix as described above and a B vector in the form $B = [0, x, 0, x, ..., 0, x]^T$ [62]. This form is only for 2×2 blocks.

Second-order modes See *Hankel Singular Values.*

Similarity Transformations A similarity transformation, T, is an invertible matrix T which is applied to the state vector and results in a new state vector: $x \rightarrow Tx$.

Some of the consequences of applying a similarity transformation are:

- Given a system with the input-output description $H = C(sI - A)^{-1}B + D$ and applying a tranformation T still results in the same input-output relation.
- Eigenvalues of a matrix are unchanged under a similarity transformation.

(Linear) State-Space System A set of linear constant-coefficient differential or difference equations represented in matrix notation as:

Continuous Time:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

Discrete Time:

$$x[k+1] = Ax[k] + Bu[k]$$
$$y[k] = Cx[k] + Du[k]$$

where $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{p \times n},$ and $D \in \mathbb{R}^{p \times m}$; $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector, and $y \in \mathbb{R}^p$ is the output vector.

For time varying systems, the $A, B, C,$ and D matrices could all be functions of time.

A periodically time varying system has system matrices of the form: $A(t+T) = A(t)$, $\forall t \in \mathbb{R}$, where T is the period. Similarly, in discrete time, $A[k+L] = A[k]$ $\forall k \in \mathbb{Z}$ where L is the period.

Unless otherwise explicitly noted, *all system descriptions are assumed to be minimal.*

Throughput The number of samples output per unit time.

Transfer Function An input/output description of a linear system given in Laplace transform notation:

$$H(s) = \frac{N(s)}{D(s)} = \frac{b_0 + b_1 s + \ldots + b_m s^m}{a_0 + a_1 s + \ldots + a_n s^n} \tag{B.11}$$

while in z-transform notation:

$$H(z) = \frac{N(z)}{D(z)} = \frac{b_0 + b_1 z + ... + b_m z^m}{a_0 + a_1 z + ... + a_n z^n} \tag{B.12}$$

Transforms The Laplace transform of the function $f(t)$ is

$$F(s) = \int_{-\infty}^{\infty} f(t) e^{-st} \, dt \tag{B.13}$$

while its Fourier transform (assuming it exists) is

$$F(j\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} \, dt \tag{B.14}$$

The (one-sided) z-transform is

$$H(z) = \sum_{k=0}^{\infty} f(k) z^{-k} \tag{B.15}$$

while its associated frequency response is

$$H(e^{j\Omega}) = \sum_{k=0}^{\infty} f(e^{j\Omega}) e^{j\Omega k} \tag{B.16}$$

Vec The vectorization operator vec applied to a matrix M of dimension $m \times n$ returns a vector of length mn whose elements are the columns of M stacked.

Zero-order-hold Given a continuous signal, $x(t)$, outputs a stairstep signal with the value $x(kT)$ for $kT \leq t < kT + T$ where T is the sampling period. The usual model assumes instantaneous transitions and ignores what happens at $t = kT$.

Appendix C

Derivations

C.1 Hadamard's Inequality

Let A be an arbitrary $n \times n$ nonsingular matrix with real elements. Then

$$(\det A)^2 \leq \prod_{i=1}^{n} \left(\sum_{k=1}^{n} a_{ik}^2 \right) \tag{C.1}$$

The term on the right side is the product of the euclidean norms of each row. Note that this is equivalent to the ith diagonal entry of $A^T A$.

$$\sum_{k=1}^{n} a_{ik}^2 = [A^T A]_{ii}$$

If A is restricted to be a positive definite, symmetric matrix, then we can continue:

$$\sum_{k=1}^{n} a_{ik}^2 = [A^2]_{ii}$$

since $A^T = A$. Thus, (C.1) reduces to

$$(\det A)^2 \leq \prod_{i=1}^{n} [A^2]_{ii}$$

$$\frac{(\det A)^2}{\prod_{i=1}^{n} [A^2]_{ii}} \leq 1$$

$$\frac{(\det A)^2}{\left(\prod_{i=1}^{n} [A]_{ii}\right)^2} \leq 1$$

$$\frac{\det A}{\prod_{i=1}^{n} [A]_{ii}} \leq 1$$

$$\left[\frac{\det A}{\prod_{i=1}^{n} [A]_{ii}}\right]^{1/2} \leq 1$$

Note that equality will hold iff $(\det A)^2 = \prod_{i=1}^{n} [A^2]_{ii}$ which will only be true if A is diagonal.

C.2 Hölder's Inequality

$$\sum_{i=1}^{u} \prod_{j=1}^{v} a_{i,j}^{r_j} \leq \prod_{j=1}^{v} \left(\sum_{i=1}^{u} a_{i,j}\right)^{r_j}$$

where $a_{i,j}$ are nonnegative real numbers for $i = 1, ..., u$ and $j = 1, ..., v$, and the r_j are nonnegative real numbers such that $\sum_{j=1}^{v} r_j = 1$.

This inequality holds as $r_i \to \infty$, or equivalently, for $\frac{1}{r_i} \to 0$.)

Let

$$n = 2; p \triangleq r_1 = 2; q \triangleq r_2 = 2;$$

$$b_j \triangleq a_{1,j}; c_j \triangleq a_{2,j};$$

Then, rewriting the general version, and noting that

$$\sum_{j=0}^{m-1}\left(\prod_{i=0}^{1} a_{i,j}\right) \leq \prod_{i=0}^{1}\left(\sum_{j=0}^{n-1} a_{i,j}^{r_i}\right)^{\frac{1}{r_i}}$$

$$\sum_{j=0}^{m-1}(a_{0,j})(a_{1,j}) \leq \left(\sum_{j=0}^{n-1} a_{0,j}^{r_0}\right)^{\frac{1}{r_0}}\left(\sum_{j=0}^{n-1} a_{1,j}^{r_1}\right)^{\frac{1}{r_1}}$$

$$\sum_{j=0}^{m-1} b_j c_j \leq \left(\sum_{j=0}^{n-1} b_j^p\right)^{\frac{1}{p}}\left(\sum_{j=0}^{n-1} c_j^q\right)^{\frac{1}{q}}$$

If $b_i, c_i \not\geq 0 \; \forall i$, then we can instead state

$$\sum_{j=0}^{m-1} |b_j||c_j| \leq \left(\sum_{j=0}^{n-1} |b_j|^p\right)^{\frac{1}{p}}\left(\sum_{j=0}^{n-1} |c_j|^q\right)^{\frac{1}{q}}$$

Finally, since

$$\sum_{j=0}^{m-1} |b_j c_j| \leq \sum_{j=0}^{m-1} |b_j||c_j|,$$

$$\sum_{j=0}^{m-1} |b_j c_j| \leq \left(\sum_{j=0}^{n-1} |b_j|^p\right)^{\frac{1}{p}}\left(\sum_{j=0}^{n-1} |c_j|^q\right)^{\frac{1}{q}}$$

If $p = q = 2$, we get the Cauchy Schwarz inequality

$$\sum_{j=0}^{m-1} |b_j c_j| \leq \left(\sum_{j=0}^{n-1} |b_j|^2\right)^{\frac{1}{2}}\left(\sum_{j=0}^{n-1} |c_j|^2\right)^{\frac{1}{2}}$$

C.3 Transfer Function Pole Sensitivity Analysis

I summarize here Kaiser's derivation of sensitivity of Direct Form realizations and their dependence on sampling rate and system order. A shortened version of the discussion here has already appeared in Section 4.3.

He starts out with the model that the digital filter design is being done to approximate a continuous-time filter with a rational transfer function. He takes the two common methods of the bilinear z-transform and the "standard" z-transformation methods.

The bilinear z-transform is defined as:

$$H(z^{-1}) = H(s)\big|_{s \to \frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}}$$

where $z = e^{sT}$.

The standard z-transformation method requires a partial fraction expansion of $H(s)$. Then, transform each partial fraction using the relationship

$$\frac{1}{s+a} \to \frac{T}{1 - e^{-aT}z^{-1}}$$

Now, this is the important part. In both cases, the resulting $H(z^{-1})$ is of the form

$$H(z^{-1}) = \frac{\sum_{i=0}^{m} b_i z^{-i}}{1 + \sum_{i=1}^{n} a_i z^{-i}} = \frac{N(z^{-1})}{D(z^{-1})} = \frac{N(z^{-1})}{\prod_{i=1}^{n}(1 - \frac{z^{-1}}{z_i})}$$

Note that only simple poles are assumed for the 'basically' low-pass transfer function $H(s)$,

$$H(s) = \frac{N(s)}{\prod_{i=1}^{n}(s - p_i)}$$

With the standard z-transformation,

$$D(z^{-1}) = \prod_{i=1}^{n}(1 - e^{p_i T} z^{-1})$$

where p_i represents the ith pole of $H(s)$ and may be complex.

For the bilinear transform,

$$
\begin{aligned}
D(z^{-1}) &= \prod_{i=1}^{n}(s-p_i)\big|_{s\rightarrow\frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}} \\
&= \prod_{i=1}^{n}\left(\frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}-p_i\right) \\
&= \prod_{i=1}^{n}\left(\frac{(1-z^{-1})}{(1+z^{-1})}-\frac{p_iT}{2}\right) \\
&= \prod_{i=1}^{n}\left((1-z^{-1})-\frac{p_iT}{2}(1+z^{-1})\right) \\
&= \prod_{i=1}^{n}\left(1-\frac{p_iT}{2}-(1+\frac{p_iT}{2})z^{-1}\right) \\
&= \prod_{i=1}^{n}\left(1-\frac{(1+\frac{p_iT}{2})}{(1-\frac{p_iT}{2})}z^{-1}\right)
\end{aligned}
$$

Note that the terms that came out from the denominator are shifted into the numerator polynomial. Assume that $\frac{1}{T}$ is greater then twice the Nyquist frequency.

Normalizing with respect to the sampling frequency,

$$\mu_i = \frac{p_iT}{\pi} = \frac{p_i}{\omega_n}$$

where ω_n is the Nyquist frequency. Assuming there is no aliasing, $|\mu_i| < 1\ \forall i$. As the sampling frequency increases, the μ_i decrease from one towards zero.

The standard and bilinear cases transform in the limit as $T \rightarrow 0$ as follows

$$\left[1-e^{p_iT}z^{-1}\right] \rightarrow \left[1-(1+\mu_i\pi)z^{-1}\right]$$

$$\left[1-\frac{(1+\frac{p_iT}{2})}{(1-\frac{p_iT}{2})}z^{-1}\right] \rightarrow \left[1-(1+\mu_i\pi)z^{-1}\right]$$

Note that $\frac{1}{1-\frac{p_iT}{2}} = \frac{1}{1-\frac{\mu_i\pi}{2}} \Rightarrow \frac{(1+\frac{p_iT}{2})}{(1-\frac{p_iT}{2})}z^{-1} \approx (1+\frac{\pi T}{2})(1+\frac{\pi T}{2}) \approx 1+\mu_i\pi$ if we drop the higher order terms. Thus, the two methods yield the same polynomial in the limit. The zeros of this polynomial (not to be confused with the zeros of $H(z^{-1})$ or $H(s)$) all tend to

$$z_i = \frac{1}{1+\mu_i\pi} \approx 1-\mu_i\pi$$

The filter will become unstable if any of the zeros of $D(z^{-1})$ move inside the unit circle. To estimate the order of the perturbation necessary to move a zero of $D(z^{-1})$ to the point $z^{-1} = 1$, consider:

$$D(z^{-1})|_{z^{-1}=1} = \prod_{i=1}^{n} \mu_i \pi = \prod_{i=1}^{n} p_i T \tag{C.2}$$

$$= 1 + \sum_{i=1}^{n} a_i z^{-i}|_{z^{-1}=1} = 1 + \sum_{i=1}^{n} a_i \tag{C.3}$$

If any of the b_i is changed by $F_0 \triangleq 1 + \sum_{j=1}^{n} b_j$, then (C.3) can be zero, which would mean a pole on the unit circle. For example, say

$$b_{i_2} = b_i - F_0$$

then

$$\begin{aligned}\sum_{j=1}^{n} b_j &= \sum_{j=1, j\neq i}^{n} b_j + b_{i_2} \\ &= \sum_{j=1, j\neq i}^{n} b_j + b_i - F_0 \\ &= 0\end{aligned}$$

Thus, $D(z^{-1})$ will have a zero on the unit circle. Of course, this bound is very crude. However, one can still make some comments regarding the relationship of wordlength and sampling rate/system order.

(C.2) implies that coefficient accuracy is affected by the sampling rate and by the system order. Thus, going from an nth order filter to a $(2n)$th order filter will require approximately twice as many bits to represent the a_i. Similarly, doubling the sampling rate for an nth order filter will require n additional bits to represent the a_i.

Interestingly, (C.3) can be interpreted as the return difference at zero frequency when the filter $H(z^{-1})$ is realized in direct form i.e. it will be very small since the filter will pass everything through. (See Figure 5-1).

Also, the problem is further aggravated as T approaches 0 since the a_i tend to approach in magnitude the binomial coefficients $\binom{n}{i}$, and tend to alternate in sign, leading to computational problems in evaluating $D(z^{-1})$ due to the differences of large numbers. This idea leads to a tighter bound on coefficient accuracy. The largest a_i is given approximately by

$$\begin{aligned}\max_i a_i &\approx \binom{n}{[n/2]} \approx \frac{2^n}{\sqrt{n}}\sqrt{\frac{2}{\pi}} \\ &\approx \frac{4}{5}\frac{2^n}{\sqrt{n}}\end{aligned}$$

which yields the bound (C.4), listed here

$$m_d \geq 1 + \lceil -\log_{10}\left(\frac{5\sqrt{n}}{2^{n+2}}\prod_{k=1}^{n} p_k T\right)\rceil \tag{C.4}$$

Trying to tighten the bound further, Kaiser computes the sensitivity of the poles to changes in the coefficients.

$$\begin{aligned}
1 + \sum_{l=1}^{n} a_i z^{-l} &= \prod_{j=1}^{n}(1 - \frac{z^{-1}}{z_j}) \\
\frac{\partial}{\partial a_k}\left(1 + \sum_{l=1}^{n} a_l z^{-l}\right) &= \frac{\partial}{\partial a_k}\prod_{j=1}^{n}(1 - \frac{z^{-1}}{z_j}) \\
z^{-k} &= \sum_{j=1}^{n}\prod_{\substack{l=1\\ l\neq j}}^{n}\left(1 - \frac{z^{-1}}{z_l}\right)\frac{\partial}{\partial a_k}\left(1 - \frac{z^{-1}}{z_j}\right) \\
z^{-k}|_{z^{-1}=z_\imath} &= \sum_{j=1}^{n}\prod_{\substack{l=1\\ l\neq j}}^{n}\left(1 - \frac{z^{-1}}{z_l}\right)\frac{\partial}{\partial a_k}\left(1 - \frac{z^{-1}}{z_j}\right)|_{z_i} \\
z_i^k &= \prod_{\substack{l=1\\ l\neq\imath}}^{n}\left(1 - \frac{z^{-1}}{z_l}\right)\left((-1)(z^{-1})\frac{(-1)}{z_i^2}\frac{\partial z_i}{\partial a_k}\right)|_{z^{-1}=z_\imath} \\
z_i^k &= \prod_{\substack{l=1\\ l\neq\imath}}^{n}\left(1 - \frac{z^{-1}}{z_l}\right)\left(\frac{1}{z_i}\frac{\partial z_i}{\partial a_k}\right) \\
\frac{\partial z_i}{\partial a_k} &= \frac{z_i^{k+1}}{\prod_{\substack{l=1\\ l\neq i}}^{n}\left(1 - \frac{z_i}{z_l}\right)}
\end{aligned}$$

which results in a total differential change of

$$dz_\imath = \sum_{k=1}^{n}\frac{\partial z_i}{\partial a_k}\ da_k$$

Note that as the poles cluster together, the term $1 - \frac{z_i}{z_l}$ will get smaller and its reciprocal will get larger, pointing to the commonly known fact that pole sensitivity increases as the poles get closer together.

Refer to [88] for more details, including a unique approach which analyzes sensitivity using the root locus method.

C.4 Matrix Derivatives

There are at least three possible ways to define the matrix derivative with respect to a matrix,

$$\frac{\partial H}{\partial A} \tag{C.5}$$

where H is a function of A, and both H and A are matrices.

The following is the development in [143]: Let $Y \in \mathbb{R}^{p \times q}$ have elements $[y_{kl}]$ whose elements are functions of the elements of $X = [x_{ij}] \in \mathbb{R}^{m \times n}$. Then, the definition of the derivative of $Y = Y(X)$ depends on how the $pqmn$ elements $\dfrac{\partial y_{kl}}{\partial x_{ij}}$ are arranged in a rectangular matrix.

The three definitions are:

1. The derivative is a partitioned matrix $[\dfrac{\partial Y}{\partial x_{ij}}]$ whose ijth partition is derived from Y by replacing each element y_{kl} with the derivatives $\dfrac{\partial y_{kl}}{\partial x_{ij}}$.

2. The derivative is a partitioned matrix $[\dfrac{\partial y_{kl}}{\partial X}]$ whose klth partition is derived from X by replacing each element x_{ij} with the derivatives $\dfrac{\partial y_{kl}}{\partial x_{ij}}$.

3. This vectorial definition, denoted $\dfrac{\partial Y^c}{\partial X^c}$, has elements $\left[\dfrac{\partial y_{kl}}{\partial x_{ij}}\right]$ arranged in the same way as the elements $[\frac{\partial y_{kl}}{\partial x_{ij}}]$ in the product vec $Y \otimes (\text{vec } X)^T$. This derivative has been ascribed the notation $\dfrac{\partial \text{vec } Y}{\partial \text{vec } 'X}$.

Pollock extensively discusses all three in [143] and describes their inter-relationships. He goes on to conclude that the "correct" definition of the matrix derivative is the third one. However, Lutz and Hakimi [109] use the second one, the most common one, to define the sensitivity function for MIMO systems.

Thus, the matrix derivative of $\dfrac{\partial Y}{\partial X}$ is an $mp \times nq$ matrix, partitioned into the $m \times n$ submatrices with the klth partition equal to $\dfrac{\partial y_{kl}}{\partial X}$.

$$\frac{\partial Y}{\partial X} = \left[\begin{array}{c|c|c|c} \dfrac{\partial y_{11}}{\partial X} & \dfrac{\partial y_{12}}{\partial X} & \cdots & \dfrac{\partial y_{1q}}{\partial X} \\ \hline \dfrac{\partial y_{21}}{\partial X} & \dfrac{\partial y_{22}}{\partial X} & \cdots & \dfrac{\partial y_{2q}}{\partial X} \\ \hline \vdots & \vdots & & \vdots \\ \hline \dfrac{\partial y_{p1}}{\partial X} & \dfrac{\partial y_{p2}}{\partial X} & \cdots & \dfrac{\partial y_{pq}}{\partial X} \end{array}\right] \tag{C.6}$$

The derivative of y_{kl} with respect to X, denoted $\dfrac{\partial y_{kl}}{\partial X}$ is a matrix of the same order as X whose ijth entry is $\dfrac{\partial y_{kl}}{\partial x_{ij}}$.

Now, to derive $\dfrac{\partial H(z)}{\partial A}$, [109] first computes $\dfrac{\partial H(z)}{\partial a_{rs}}$ and then, using Graham's "First Transformation Principle" [66], computes $\dfrac{\partial [H(z)]_{ij}}{\partial A}$. I will present the discussion of Graham's "First Transformation Principle" presented in Lutz and Hakimi's paper.

Consider a matrix product of the form $Y = MXN$ where $X = [x_{rs}]$ is of order $m \times n$, $Y = [y_{ij}]$ is of order $l \times q$, and M and N are matrices compatible with X and are independent of X. Then,

$$\frac{\partial y_{ij}}{\partial X} = M^T E_{ij} N^T$$

where $E_{ij} = e_i e_j^T$ is an elementary matrix of the same order as the matrix Y.

Also,

$$\frac{\partial Y}{\partial x_{rs}} = \frac{\partial (MXN)}{\partial x_{rs}} = M E_{rs} N$$

where E_{rs} is an elementary matrix of the same order as the matrix X. Graham's "First Transformation Principle" states that $\dfrac{\partial y_{ij}}{\partial X}$ is a transformation of $\dfrac{\partial Y}{\partial x_{rs}}$ and vice versa. For example, to obtain $\dfrac{\partial y_{ij}}{\partial X}$ from $\dfrac{\partial Y}{\partial x_{rs}}$, replace M by M^T, N by N^T, and E_{rs} by E_{ij} (changing the size of the elementary matrix if necessary).

Applying this idea yields

$$\begin{aligned}\frac{\partial H(z)}{\partial A} &= \left[\begin{array}{c|c|c|c} G_1F_1^T & G_1F_2^T & \cdots & G_1F_l^T \\ \hline G_2F_1^T & G_2F_2^T & \cdots & G_2F_l^T \\ \hline \vdots & \vdots & & \vdots \\ \hline G_mF_1^T & G_1F_2^T & \cdots & G_mF_l^T \end{array}\right] \\ &= (\text{vec } G)(\text{vec } F)^T\end{aligned}$$

where G_i, F_j represent the ith and jth column of G and F respectively.

C.5 Using the Cauchy-Schwarz Inequality to bound $M_{L_{12}}$

Applying the Cauchy-Schwarz inequality to the first term, call it M_1, in both $M_{L_{12}}$ measures,

$$\begin{aligned}M_1 &= \left(\frac{1}{2\pi}\int_0^{2\pi} \|G\|_F\|F\|_F \; d\omega\right)^2 \\ &\leq \left(\frac{1}{2\pi}\int_0^{2\pi} \|G\|_F^2 \; d\omega\right)\left(\frac{1}{2\pi}\int_0^{2\pi} \|F\|_F^2 \; d\omega\right) \\ &= \text{tr}\,(W_o)\text{tr}\,(W_c)\end{aligned}$$

A slightly more detailed explanation:

$$\left[\frac{\partial H(z)}{\partial A}\right] = S_{ij}(z) = G_i(z)F_j(z)$$

Applying the 1-norm,

$$\begin{aligned}
\|S(z)\|_1^2 &= \left(\frac{1}{2\pi}\int_0^{2\pi} \|S(e^{j\omega})\|_F \; d\omega\right)^2 \\
\text{where} \quad & \|S(e^{j\omega})\|_F = \sum_i \sum_j \|S_{ij}(e^{j\theta})\|^2 \\
&= \left(\frac{1}{2\pi}\int_0^{2\pi} \|G^T(e^{j\omega})F^T(e^{j\omega})\|_F \; d\omega\right)^2 \\
&= \left(\frac{1}{2\pi}\int_0^{2\pi} [G(e^{-j\omega})G^T(e^{j\omega})F(e^{-j\omega})F^T(e^{j\omega})]^{1/2} \; d\omega\right)^2 \\
&\leq \left(\frac{1}{2\pi}\int_0^{2\pi} G(e^{-j\omega})G^T(e^{j\omega}) \; d\omega\right) \\
&\quad \left(\frac{1}{2\pi}\int_0^{2\pi} F(e^{-j\omega})F^T(e^{j\omega}) \; d\omega\right) \\
&= \left\|\frac{\partial H(z)}{\partial B}\right\|_2^2 \left\|\frac{\partial H(z)}{\partial C}\right\|_2^2
\end{aligned}$$

An alternative way to reach the same result:

$$\begin{aligned}
\|S_{ij}(e^{j\theta})\|_1^2 &= \left(\frac{1}{2\pi}\int_0^{2\pi} \|S_{ij}(e^{j\omega})\|_F \; d\omega\right)^2 \\
&= \left(\frac{1}{2\pi}\int_0^{2\pi} \|G_i(e^{j\omega})F_j(e^{j\omega})\|_F \; d\omega\right)^2 \\
&\leq \left(\frac{1}{2\pi}\int_0^{2\pi} \|G_i(e^{j\omega})\|^2 \; d\omega\right)^2 \\
&\quad \left(\frac{1}{2\pi}\int_0^{2\pi} \|F_j(e^{j\omega})\|^2 \; d\omega\right)^2 \\
&= [W_c]_{ii}[W_o]_{jj}
\end{aligned}$$

Summing both sides over all i, j,

$$\sum_i \sum_j \|S_{ij}(e^{j\theta})\|_1^2 \leq \sum_i \sum_j [W_o]_{ii}[W_c]_{jj} = Tr(W_o)Tr(W_c)$$

Bibliography

[1] J. W. Adams and A. N. Willson, Jr. Some efficient digital prefilter structures. *IEEE Transactions on Circuits and Systems*, CAS-31(3):260–265, March 1984.

[2] R. C. Agarwal and C. S. Burrus. New recursive digital filter structures having very low sensitivity and roundoff noise. *IEEE Transactions on Circuits and Systems*, CAS-22(12):921–927, December 1975.

[3] G. Amit and U. Shaked. Small roundoff noise realization of fixed-point digital filters and controllers. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-36(6):880–891, June 1988.

[4] A. Anuff and C. Y. Kao. Comments on "on the evaluation of roundoff noise in digital filters". *IEEE Transactions on Circuits and Systems*, CAS-23(9):573, September 1976.

[5] U. Appel. Bounds on second-order digital filter limit cycles. *IEEE Transactions on Circuits and Systems*, CAS-22(7):630–632, July 1975.

[6] M. Araki. Recent developments in digital control theory. In *IFAC 12th Triennial World Congress*, 1993.

[7] E. Avenhaus. On the design of digital filters with coefficients of limited wordlength. *IEEE Transactions on Audio Electroacoustics*, 1972. Also in [132].

[8] B. A. Bamieh and J. B. Pearson, Jr. A general framework for linear periodic systems with applications to $\mathcal{H}^\infty$ sampled-data control. *IEEE Transactions on Automatic Control*, 37(4):418–435, April 1992.

[9] C. W. Barnes. Roundoff noise and overflow in normal digital filters. *IEEE Transactions on Circuits and Systems*, CAS-26(3):154–159, March 1979.

[10] C. W. Barnes. Error feedback in normal realization of recursive digital filters. *IEEE Transactions on Circuits and Systems*, CAS-28(1):72–75, January 1981.

[11] C. W. Barnes. Computationally efficient second-order digital filter sections with low roundoff noise gain. *IEEE Transactions on Circuits and Systems*, CAS-31(10):841–847, October 1984.

[12] C. W. Barnes. On the design of optimal state-space realizations of second-order digital filters. *IEEE Transactions on Circuits and Systems*, CAS-31(7):602–608, July 1984.

[13] C. W. Barnes and T. Miyawaki. Roundoff noise invariants in normal digital filters. *IEEE Transactions on Circuits and Systems*, CAS-29(4):251–256, April 1982.

[14] C. W. Barnes and S. Shinnaka. Finite word effects in block-state realizations of fixed point digital filters. *IEEE Transactions on Circuits and Systems*, CAS-27:345–349, May 1980.

[15] C. W. Barnes, B. N. Tran, and S. H. Leung. On the statistics of fixed-point roundoff error. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33(3):595–606, June 1985.

[16] P. H. Bauer and L.-J. Leclerc. A computer-aided test for the absence of limit cycles in fixed-point digital filters. *IEEE Transactions on Signal Processing*, 39(11):2400–2410, November 1991.

[17] P. H. Bauer and K. Premaratne. Limit cycles in delta-operator formulated 1-d and m-d discrete-time systems with fixed-point arithmetic. *IEEE Transactions on Circuits and Systems-I:Fundamental Theory and Applications*, 44(6):529–537, June 1997.

[18] A. A. L. Beex and V. E. Debrunner. Reduced sensitivities of direct form digital (sub) filter structures by increasing system order. *IEEE Transactions on Circuits and Systems*, 36(3):438–442, March 1989.

[19] S. E. Belter and S. C. Bass. Computer-aided analysis and design of digital filters with arbitrary topology. *IEEE Transactions on Circuits and Systems*, CAS-22(10):810–819, October 1975.

[20] W. R. Bennett. Spectra of quantized signals. *Bell Systems Technical Journal*, 27:446–472, July 1948.

[21] A. G. Bolton. A two's complement overflow limit cycle free digital filter structure. *IEEE Transactions on Circuits and Systems*, CAS-31(12):1045–1046, December 1984.

[22] B. W. Bomar. New second-order state-space structures for realizing low roundoff noise digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1985.

[23] B. W. Bomar and J. C. Hung. Minimum roundoff noise digital filters with some power-of-two coefficients. *IEEE Transactions on Circuits and Systems*, CAS-31(10):833–840, October 1984.

[24] N. K. Bose. *Digital filters : theory and applications.* Elsevier Science Publishing Co., New York, 1985.

[25] L. T. Bruton and D. A. Vaughan-Pope. Synthesis of digital ladder filters from lc filters. *IEEE Transactions on Circuits and Systems*, CAS-23(6):395–402, June 1976.

[26] C. S. Burrus. Block implementation of digital filters. *IEEE Transactions of Circuit Theory*, CT-18:697–701, November 1971.

[27] C. S. Burrus. Block realization of digital filters. *IEEE Transactions of Audio Electronics*, AU-20:230–235, October 1972.

[28] H. Butterweck, J. Ritzerfeld, and M. Werter. Finite wordlength effects in digital filters: A review. Technical Report EUT 88-E-205, Eindhoven University of Technology, October 1988.

[29] H.-J. Butterweck, J. Ritzerfeld, and M. Werter. Finite wordlength effects in digital filters. *Archiv der elektrischen Ubertragung*, 43(2):76–89, 1989.

[30] H. J. Butterweck, A. C. P. Van Meer, and G. Verkrooost. New second-order digital filter sections without limit cycles. *IEEE Transactions on Circuits and Systems*, CAS-31(2):141–146, February 1984.

[31] F. Catthoor, H. De Man, and J. Vandewalle. Simulated-annealing-based optimization of coefficient and data word-lengths in digital filters. *International Journal of Circuit Theory and Applications*, 16:371–390, 1988.

[32] T. Çiloğlu and Z. Ünver. A new approach to discrete coefficient FIR digital filter design by simulated annealing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 101–104, New York, 1993. IEEE, IEEE.

[33] D. S. K. Chan. *Theory and implementation of multidimensional discrete systems for signal processing.* PhD thesis, Massachusetts Institute of Technology, 1978.

[34] K. Chang and W. G. Bliss. Finite word-length effects of pipelined recursive digital filters. *IEEE Transactions on Signal Processing*, 42(8):1983–1995, August 1994.

[35] K. Chang and W. G. Bliss. Reply to "comments on 'finite word-length effects of pipelined recursive digital filters'". *IEEE Transactions on Signal Processing*, 43(12):3032, December 1995.

[36] T. Chen and B. Francis. *Optimal sampled-data control systems.* Springer-Verlag, London, 1995.

[37] T. Chen and B. A. Francis. $\mathcal{H}_2$-optimal sampled-data control. *IEEE Transactions on Automatic Control*, 36(4):387–397, April 1991.

[38] T. A. C. M. Claasen, W. F. G. Mecklenbråuker, and J. B. H. Peek. On the stability of the forced response of digital filters with overflow nonlinearities. *IEEE Transactions on Circuits and Systems*, CAS-22(8):692–696, August 1975.

[39] T. A. C. M. Claasen, W. F. G. Mecklenbråuker, and J. B. H. Peek. Quantization noise analysis for fixed-point digital filters using magnitude truncation for quantization. *IEEE Transactions on Circuits and Systems*, CAS-22(11):887–895, November 1975.

[40] R. E. Crochiere. A new statistical approach to the coefficient word length problem for digital filters. *IEEE Transactions on Circuits and Systems*, CAS-22(3):190–196, March 1975.

[41] M. Dahleh, M. A. Dahleh, and G. C. Verghese. Course notes for 6.241. Part of a set under development.

[42] A. G. Dempster and M. D. Macleod. Use of minimum-adder multiplier blocks in FIR digital filters. *IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing*, 42(9):569–577, September 1995.

[43] A. G. Dempster and M. D. Macleod. Comparison of fixed-point FIR digital filter design techniques. *IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing*, 44(7):591–593, July 1997.

[44] P. S. R. Diniz and A. Antoniou. On the elimination of constant-input limit cycles in digital filters. *IEEE Transactions on Circuits and Systems*, CAS-31(7):670–671, July 1984.

[45] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback control theory.* Macmillan Publishing Co., New York, 1992.

[46] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard $\mathcal{H}_{\in}$ and $\mathcal{H}_{\infty}$ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, August 1989.

[47] G. E. Dullerud and B. A. Francis. $\mathcal{L}_1$ analysis and design of sampled-data systems. *IEEE Transactions on Automatic Control*, 37(4):436–446, April 1992.

[48] H. S. El-Ghoroury and S. C. Gupta. Wave digital filter structures with variable frequency characteristics. *IEEE Transactions on Circuits and Systems*, CAS-23(10):624–630, October 1976.

[49] A. Emami-Naeini and P. Van Dooren. Computation of zeros of linear multivariable systems. *Automatica*, 18(4):415–430, 1982.

[50] J. Fadavi-Ardekani, S. K. Mitra, and B. D. O. Anderson. Extended state-space model of discrete-time dynamical systems. *IEEE Transactions on Circuits and Systems*, CAS-29(8):547–556, August 1982.

[51] A. Fettweis. On the evaluation of roundoff noise in digital filters. *IEEE Transactions on Circuits and Systems*, CAS-22(11):896–897, November 1975.

[52] A. Fettweis. Wave digital filters: theory and practice. *Proceedings of the IEEE*, 74(2):270–327, February 1986.

[53] A. Feuer and G. C. Goodwin. *Sampling in digital signal processing and control.* Systems and Control. Birkhauser, Boston, 1996.

[54] I. J. Fialho and T. T. Georgiou. On stability and performance of sampled-data systems subject to wordlength constraint. In *Proceedings of the 32nd Conference on Decision and Control*, volume 1, pages 309–314, New York, 1993. IEEE, IEEE.

[55] I. J. Fialho and T. T. Georgiou. On stability and performance of sampled-data systems subject to wordlength constraint. *IEEE Transactions on Automatic Control*, 39(12):2476–2481, December 1994.

[56] A. M. Fink. A bound on quantization errors in second-order digital filters with complex poles that is tight for small θ. *IEEE Transactions on Circuits and Systems*, CAS-23(5):325–326, May 1976.

[57] H. Fujimoto and A. Kawamura. New digital redesign method by n-delay control and its application. In *Proceedings of the Power Conversion Conference-Nagaoka 1997*, volume 1, pages 525–530, New York, 1997. IEEE, IEEE Press.

[58] H. Fujimoto and A. Kawamura. Generalized digital redesign method by n-delay control and its application to motion control. In *To appear in Power Electronics Specialist Conference*, May 1998.

[59] H. Fujimoto, A. Kawamura, and M. Tomizuka. Proposal of generalized digital redesign method in use of n-delay control. In *Proceedings of the American Control Conference*, pages 3200–3204, 1997.

[60] P. Gentili, F. Piazza, and A. Uncini. Improved power-of-two sharpening filter design by genetic algorithm. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1375–1378, New York, 1996. IEEE, IEEE.

[61] A. P. Gerheim. Calculation of quantization noise at non-storage nodes. *IEEE Transactions on Circuits and Systems*, CAS-31(12):1054–1055, December 1984.

[62] M. Gevers and G. Li. *Parametrizations in Control, Estimation and Filtering Problems:Accuracy Aspects.* Communications and Control Engineering Series. Springer-Verlag, London, 1993.

[63] B. Gold and K. L. Jordan. A note on digital filter synthesis. *Proceedings of the IEEE*, 65:1717–1718, October 1968.

[64] G. C. Goodwin, R. H. Middleton, and H. V. Poor. High-speed digital signal processing and control. *Proceedings of the IEEE*, 80(2):240–259, February 1992.

[65] G. Goossens, J. V. Praet, D. Lanneer, W. Geurts, A. Kifli, C. Liem, and P. G. Paulin. Embedded software in real-time signal processing systems: Design technologies. In *Proceedings of the IEEE*, volume 85, pages 436–454. IEEE, March 1997.

[66] A. Graham. *Kronecker Products and Matrix Calculus with Applications.* Halsted, div. of Wiley, 1981.

[67] R. M. Gray. Quantization noise spectra. *IEEE Transactions on Information Theory*, 36(6):1220–1244, November 1990.

[68] H. Hanselmann. Implementation of digital controllers – a survey. *Automatica*, 23(1):7–32, 1987.

[69] M. W. Hauser. Principles of oversampling a/d conversion. *Journal of the Audio Engineering Society*, 39(1/2):3–26, January/February 1991.

[70] W. E. Higgins and D. C. Munson, Jr. Noise reduction strategies for digital filters:error spectrum shaping versus the optimal linear state-space formulation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30(6):963–973, December 1982.

[71] W. E. Higgins and D. C. Munson, Jr. Optimal and suboptimal error spectrum shaping for cascade-form digital filters. *IEEE Transactions on Circuits and Systems*, CAS-31(5):429–437, May 1984.

[72] K. Hwang. *Computer Arithmetic.* Wiley, New York, 1979.

[73] S. Hwang. Minimum uncorrelated unit noise in state-space digital filtering. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-25(8):273–281, August 1977.

[74] S. Y. Hwang. Comments on "Reducation of roundoff noise in wave digital filters". *IEEE Transactions on Circuits and Systems*, CAS-22(9), September 1975.

[75] IEEE arithmetic. http://www.medusa.uni-bremen.de/intern/spro/common-tools/numerical_comp_guide/ncg_math.doc.html, 1985.

[76] R. Ishii. Realization of a digital filter using a variation of partial fraction expansion. *IEEE Transactions on Circuits and Systems*, CAS-23(3):178–181, March 1976.

[77] M. Iwatsuki, M. Kawamata, and T. Higuchi. Statistical sensitivity and minimum sensitivity structures with fewer coefficients in discrete time linear systems. *IEEE Transactions on Circuits and Systems*, 37(1):72–80, January 1989.

[78] L. B. Jackson. Roundoff-noise analysis for fixed-point digital filters realized in cascade or parallel form. *IEEE Transactions on Audio and Electroacoustics*, AU-18:107–122, June 1970. Also in [145].

[79] L. B. Jackson. Roundoff noise bounds derived from coefficient sensitivities for digital filters. *IEEE Transactions on Circuits and Systems*, CAS-23(8):481–485, August 1976.

[80] L. B. Jackson. Comments on 'finite word-length effects of pipelined recursive digital filters'. *IEEE Transactions on Signal Processing*, 43(12):3031, December 1995.

[81] L. B. Jackson. *Digital filters and signal processing : with MATLAB exercises.* Kluwer Academic Publishers, Boston, third edition, 1996.

[82] L. B. Jackson, A. G. Lindgren, and Y. Kim. Optimal synthesis of second-order state-space structures for digital filters. *IEEE Transactions on Circuits and Systems*, CAS-26(3):149–153, March 1979.

[83] Y. Jang and S. P. Kim. Block digital filter structures and their finite precision responses. *IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing*, 43(7):495–506, July 1996.

[84] W. K. Jenkins and B. J. Leon. An analysis of quantization error in digital filters based on interval algebras. *IEEE Transactions on Circuits and Systems*, CAS-22(3):223–232, March 1975.

[85] Z. Jiang and A. N. J. Willson. Design and implementation of efficient pipelined iir digital filters. *IEEE Transactions on Signal Processing*, 43:579–590, March 1995.

[86] Z. Jiang and A. N. J. Willson. A pipelined/interleaved iir digital filter architecture. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 2217–2219. IEEE, 1997.

[87] J. F. Kaiser. Design methods for sampled data filters. In *Proceedings of the First Annual Allerton Conference on Circuit and System Theory*, pages 221–236, 1963.

[88] J. F. Kaiser. Some practical considerations in the realization of linear digital filters. In *Proceedings of the Third Annual Allerton Conference on Circuit and System Theory*, pages 621–633, 1965.

[89] C. Kasarabada and B. A. Shenoi. On the zeros of polynomials used in digital filters. *IEEE Transactions on Circuits and Systems*, CAS-23(7):423–429, July 1976.

[90] P. Katz. *Digital control using microprocessors.* Prentice Hall International, Englewood Cliffs, N.J., 1981.

[91] M. Kawamata and T. Higuchi. A unified approach to the optimal synthesis of fixed-point state-space digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33(4):911–920, August 1985.

[92] J. P. Keller and B. D. O. Anderson. A new approach to the discretization of continuous-time controllers. *IEEE Transactions on Automatic Control*, 37(2):214–223, February 1992.

[93] J. B. Knowles and R. Edwards. Effect of a finite-word-length computer in a sampled-data feedback system. In *Proceedings of the IEE*, volume 112, pages 1197–1207. IEE, June 1965.

[94] J. B. Knowles and E. M. Olcayto. Coefficient accuracy and digital filter response. *IEEE Transactions on Circuit Theory*, CT-15:31–41, March 1968. Also in [132].

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

[96] T. I. Laakso and I. O. Hartimo. Noise reduction in recursive digital filters using higher-order error feedback. *IEEE Transactions on Signal Processing*, 40(5):1096–1107, May 1992.

[97] P. Lancaster and M. Tismenetsky. *The theory of matrices: with applications.* Computer science and applied mathematics. Academic Press, second edition, 1985.

[98] L.-J. Leclerc and P. H. Bauer. New criteria for asymptotic stability of one- and multidimensional state-space digital filters in fixed-point arithmetic. *IEEE Transactions on Signal Processing*, 42(1):46–53, January 1994.

[99] G. Li. On pole and zero sensitivity of linear systems. *IEEE Transactions on Circuits and Systems-I:Fundamental Theory and Applications*, 44(7):583–590, July 1997.

[100] G. Li. On the structure of digital controllers with finite word length consideration. *IEEE Transactions on Automatic Control*, 43(5):689–693, May 1998.

[101] G. Li and M. Gevers. On the structure of digital controllers in sampled-data systems with fwl consideration. In *Proceedings of the 35th IEEE Conference on Decision and Control*, volume 1, pages 919–920. IEEE, IEEE, 1996.

[102] Y. C. Lim and S. R. Parker. On the synthesis of lattice parameter digital filters. *IEEE Transactions on Circuits and Systems*, CAS-31(7):593–601, July 1984.

[103] B. Liu. Effect of finite word length on the accuracy of digital filters – a review. *IEEE Transactions on Circuit Theory*, CT-18:670–677, November 1971.

[104] B. Liu and A. Peled. Heuristic optimization of the cascade realization of fixed-point digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(5):464–473, October 1975.

[105] K. Liu, R. E. Skelton, and K. Grigoriadis. Optimal controllers for finite wordlength implementation. *IEEE Transactions on Automatic Control*, 37(9):1294–1304, September 1992.

[106] L. Liu, T. Yoshino, S. Sprouse, and R. Jain. An interleaved/retimed architecture for the lattice wave digital filter. *IEEE Transactions on Circuits and Systems*, 38(3):344–347, March 1991.

[107] P.-H. Lo and Y.-C. Jenq. Minimum sensitivity realization of second order recursive digital filter. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30(6):930–937, December 1982.

[108] J. L. Long and T. N. Trick. A note on absolute bounds on quantization errors in fixed-point implementations of digital filters. *IEEE Transactions on Circuits and Systems*, CAS-22(6):567–570, June 1975.

[109] W. J. Lutz and S. L. Hakimi. Design of multi-input multi-output systems with minimum sensitivity. *IEEE Transactions on Circuits and Systems*, 35(9):1114–1121, September 1988.

[110] A. G. Madievski and B. D. O. Anderson. Sampled-data controller reduction procedure. *IEEE Transactions on Automatic Control*, 40(11):1922–1926, November 1995.

[111] A. G. Madievski, B. D. O. Anderson, and M. Gevers. Optimum realizations of sampled-data controllers for fwl sensitivity minimization. *Automatica*, 31(3):367–379, March 1995.

[112] G. A. Maria and M. M. Fahmy. Limit cycle oscillations in a cascade of first- and second-order digital sections. *IEEE Transactions on Circuits and Systems*, CAS-22(2):131–134, February 1975.

[113] J. D. Markel and A. H. Gray, Jr. Fixed-point implementation algorithms for a class of orthogonal polynomial filter structures. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(5):486–494, October 1975.

[114] J. D. Markel and A. H. Gray, Jr. Roundoff noise characteristics of a class of orthogonal polynomial structures. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(5):473–486, October 1975.

[115] R. A. Meyer and C. S. Burrus. A unified analysis of multirate periodically time varying digital filters. *IEEE Transactions on Circuits and Systems*, CAS-22:162–168, March 1975.

[116] R. A. Meyer and C. S. Burrus. Design and implementation of multirate digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24:55–58, February 1976.

[117] R. H. Middleton and G. C. Goodwin. *Digital Control and Estimation : A Unified Approach.* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[118] S. K. Mitra and R. Gnanesekaran. Block implementation of recursive digital filters – new structures and properties. *IEEE Transactions on Circuits and Systems*, CAS-25:200–207, April 1978.

[119] P. Moroney. *Issues in the implementation of digital feedback compensators.* MIT Press, Cambridge, Mass., 1983.

[120] P. Moroney, A. S. Willsky, and P. K. Houpt. The digital implementation of control compensators: The coefficient wordlength issue. *IEEE Transactions on Automatic Control*, AC-25(4):621–630, August 1980.

[121] C. T. Mullis and R. A. Roberts. Roundoff noise in digital filters: Frequency transformations and invariants. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24:538–550, December 1976.

[122] C. T. Mullis and R. A. Roberts. Synthesis of minimum roundoff noise fixed point digital filters. *IEEE Transactions on Circuits and Systems*, CAS-23(9):551–562, September 1976.

[123] C. T. Mullis and R. A. Roberts. An interpretation of error spectrum shaping in digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30(6):1013–1015, December 1982.

[124] D. C. Munson and B. Liu. Narrow-band recursive filters with error spectrum shaping. *IEEE Transactions on Circuits and Systems*, CAS-28(2):160–163, February 1981.

[125] D. C. Munson, Jr., J. H. Strickland, Jr., and T. P. Walker. Maximum amplitude zero-input limit cycles in digital filters. *IEEE Transactions on Circuits and Systems*, CAS-31(3):266–275, March 1984.

[126] K. Natarajan and S. Sivakumar. Design of a fixed-point digital filter for state estimation for ups applications. In *IEEE 4th Workshop on Computers in Power Electronics*, pages 221–225. IEEE, IEEE, 1994.

[127] S. Nishimura, K. Hirano, and R. N. Pal. A new class of very low sensitivity and low roundoff noise recursive digital filter structures. *IEEE Transactions on Circuits and Systems*, CAS-28(12):1152–1157, December 1981.

[128] K. J. øAstrom. *Introduction to Stochastic Control Theory*, volume 70 of *Mathematics in science and engineering*. Academic Press, New York, 1970.

[129] K. J. øAstrom and B. Wittenmark. *Computer Controlled Systems : Theory and Design*. Prentice Hall, Englewood Cliffs, New Jersey, 1984.

[130] A. V. Oppenheim and R. W. Schafer. *Discrete-time signal processing.* Prentice-Hall signal processing series. Prentice Hall, Englewood Cliffs, N.J., 1989.

[131] A. V. Oppenheim and I. T. Willsky, Alan S. with Young. *Signals and Systems.* Prentice-Hall signal processing series. Prentice Hall, Englewood Cliffs, New Jersey, first edition, 1983.

[132] A. Oppenhiem and L. R. Rabiner, editors. *Selected papers in digital signal processing II.* IEEE Press, 1976.

[133] G. Orlandi and G. Martinelli. Low-sensitivity recursive digital filters obtained via the delay replacement. *IEEE Transactions on Circuits and Systems*, CAS-31(7):654–657, July 1984.

[134] K. K. Parhi. Finite word effects in pipelined recursive filters. *IEEE Transactions on Signal Processing*, 39(6):1450–1454, June 1991.

[135] K. K. Parhi and D. G. Messerschmitt. Pipeline interleaving and parallelism in recursive digital filters – part i: Pipelining using scattered look-ahead and decomposition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):1099–1117, July 1989.

[136] K. K. Parhi and D. G. Messerschmitt. Pipeline interleaving and parallelism in recursive digital filters – part ii: Pipelined incremental block filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):1118–1134, July 1989.

[137] S. R. Parker and P. E. Girard. Correlated noise due to roundoff in fixed point digital filters. *IEEE Transactions on Circuits and Systems*, CAS-23(4):204–211, April 1976.

[138] S. R. Parker and S. Yakowitz. A general method for caculating quantization error bounds due to roundoff in multivariable digital variables. *IEEE Transactions on Circuits and Systems*, CAS-22(6):570–572, June 1975.

[139] R. V. Patel, A. J. Laub, and P. M. van Dooren, editors. *Numerical linear algebra techniques for systems and control.* IEEE Press, Piscataway, NJ, 1994.

[140] P. G. Paulin, C. Liem, M. Cornero, F. Naçabal, and G. Goossens. Embedded software in real-time signal processing systems: Application and architecture trends. In *Proceedings of the IEEE*, volume 85, pages 419–435. IEEE, March 1997.

[141] J. Perkins, U. Helmke, and J. Moore. Balanced realizations via gradient flow techniques. *Systems and Control Letters*, 14(5):369–380, 1990.

[142] M. E. Polites. Ideal state reconstructor for deterministic digital control systems. *International Journal of Control*, 49(6):2001–2011, 1989.

[143] D. Pollock. Tensor products and matrix differential calculus. *Linear Algebra and Its Applications*, (67):169–193, 1985.

[144] K. Premaratne, E. Kulasekere, P. Bauer, and L. Leclerc. An exhaustive search algorithm for checking limit cycle behavior of digital filters. In *IEEE International Symposium on Circuits and Systems*, volume 3, pages 2035–2038, New York, 1995. IEEE, IEEE.

[145] L. R. Rabiner and C. M. Rader. *Digital Signal Processing.* IEEE Press, 1972.

[146] D. V. B. Rao. Analysis of coefficient quantization errors in state-space digital filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34(1):131–139, February 1986.

[147] S. K. Rao and T. Kailath. Orthogonal digital filters for vlsi implementation. *IEEE Transactions on Circuits and Systems*, CAS-31(11):933–945, November 1984.

[148] K. S. Rattan. Digitalization of existing continuous control systems. *IEEE Transactions on Automatic Control*, AC-29(3):282–285, March 1984.

[149] K. S. Rattan and H. H. Yeh. Discretizing continuous-data control systems. *Computer Aided Design*, 10:299–306, September 1978.

[150] R. L. Reng, K. Schwarz, and H. W. Schuessler. Analyzing finite-wordlength effects in block state space filters. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 85–88, New York, 1993. IEEE, IEEE.

[151] R. A. Roberts and C. T. Mullis. *Digital Signal Processing.* Addison-Wesley Series in Electrical Engineering: Digital Signal Processing. Addison-Wesley, Reading, Massachusetts, 1987.

[152] M. A. Rotea and D. Williamson. Optimal realizations of finite wordlength digital filters and controllers. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 42(2):61–72, February 1995.

[153] M. Safonov, E. Jonckheere, M. Verma, and D. Limebeer. Synthesis of positive real multivariable feedback systems. *International Journal of Control*, 45(3):817–842, March 1987.

[154] L.-S. Shieh, J.-L. Zhang, and J. W. Sunkel. A new approach to the digital redesign of continuous-time controllers. *Control-Theory and Advanced Technology*, 8(1):37–57, March 1992.

[155] L.-S. Shieh, X.-M. Zhao, and J.-L. Zhang. Locally optimal-digital redesign of continuous systems. *IEEE Transactions on Industrial Electronics*, 36(4):511–515, November 1989.

[156] S. R. Signell, T. G. Kouyoumdjiev, K. H. Mossberg, and C. G. L. Harnefors. Design and analysis of bilinear digital ladder filters. *IEEE Transactions on Circuits and Systems-I:Fundamental Theory and Applications*, 43(2):69–81, February 1996.

[157] R. E. Skelton and D. A. Wagie. Minimal root sensitivity in linear systems. *Journal of Guidance and Control*, 7:570–574, September-October 1984.

[158] M. L. Smith and B. W. Bomar. An algorithm for constrained roundoff noise minimization in digital filters with application to two-dimensional filters. *IEEE Transactions on Circuits and Systems*, 35(11):1359–1368, November 1988.

[159] M. A. Soderstrand and B. Sinha. A pipelined recursive residue number system digital filter. *IEEE Transactions on Circuits and Systems*, CAS-31(4):415–417, April 1984.

[160] S. Sridharan and D. Williamson. Comments on "suppresion of limit cycles in digital filters designed with one magnitude-truncation quantizer". *IEEE Transactions on Circuits and Systems*, CAS-31(2):235–236, February 1984.

[161] A. Sripad and D. L. Snyder. A necessary and sufficient condition for quantization errors to be uniform and white. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-25:442–448, October 1977.

[162] K. Steiglitz. The equivalence of digital and analog signal processing. *Information and Control*, 8:455–467, 1965.

[163] G. W. Stewart and J.-g. Sun. *Matrix Perturbation Theory.* Academic Press, 1990.

[164] A. A. Stoorvogel. The robust $\mathcal{H}_{\in}$ control problem: A worst-case design. *IEEE Transactions on Automatic Control*, 38(9):1358–1370, September 1993.

[165] W. Sun, K. M. Nagpal, and P. P. Khargonekar. $\mathcal{H}_{\infty}$ control and filtering for sampled-data systems. *IEEE Transactions on Automatic Control*, 38(8):1162–1175, August 1993.

[166] W. Sung and K.-I. Kum. Simulation-based word-length optimization method for fixed-point digital signal processing systems. *IEEE Transactions on Signal Processing*, 43(12):3087–3090, December 1995.

[167] E. E. Swartzlander, Jr., editor. *Computer Arithmetic*, volume 1,2 of *IEEE Computer Society Press tutorial.* IEEE Computer Society Press, Los Alamitos, California, 1990.

[168] D. Tabak. Digitalization of control systems. *Computer Aided Design*, 3(2):13–18, 1971.

[169] V. Tavşanoğlu and L. Thiele. Optimal design of state-space digital filters by simultaneous minimization of sensitivity and roundoff noise. *IEEE Transactions on Circuits and Systems*, CAS-31(10):884–888, October 1984.

[170] A. Tawfik, P. Agathoklis, and F. El-Guibaly. A tool for analyzing finite wordlength effects in fixed-point digital filter implementations. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, volume 1, pages 116–119, New York, 1993. IEEE, IEEE.

[171] A. Tawfik, P. Agathoklis, and F. El-Guibaly. New low roundoff noise realizations of second-order digital filter sections. *IEEE Transactions on Signal Processing*, 43(5):1255–1258, May 1995.

[172] L. Thiele. Design of sensitivity and roundoff noise optimal state-space discrete systems. *International Journal for Circuit Theory Applications*, 12:39–46, 1984.

[173] L. Thiele. On the sensitivity of linear state-space systems. *IEEE Transactions on Circuits and Systems*, CAS-33(5):502–510, May 1986.

[174] I. Tokaji and C. W. Barnes. Minimum unit noise gain in non-minimal state-space realizations of digital filters. *IEEE Transactions on Circuits and Systems*, 35(4):455–457, April 1988.

[175] M. Toy and P. M. Chirlian. Low multiplier coefficient sensitivity block digital filters. *IEEE Transactions on Circuits and Systems*, CAS-31(12):993–1001, December 1984.

[176] J. S. Tsai, L. S. Shieh, J. L. Zhang, and N. P. Coleman. Digital redesign of pseudo-continuous-time suboptimal regulators for large-scale discrete systems. *Control Theory and Advanced Technology*, 5(1):37–65, 1989.

[177] J. S. H. Tsai, M. S. Chen, and F. C. Kung. Optimal design and optimal digital redesign for continuous-time input time-delay systems. *Control-Theory and Advanced Technology*, 8(2):315–340, June 1992.

[178] Z. Ünver and K. Abdullah. A tighter practical bound on quantization errors in second-order digital filters with complex conjugate poles. *IEEE Transactions on Circuits and Systems*, CAS-22(7), July 1975.

[179] P. Van Dooren and B. Wyman. *Linear algebra for control theory*, volume 62 of *IMA volumes in mathematics and its applications.* Springer-Verlag, New York, 1994.

[180] G. Verkrooost and G. J. Bosscha. On the measurement of quantization noise in digital filters. *IEEE Transactions on Circuits and Systems*, CAS-31(2):222–223, February 1984.

[181] H. B. Voelcker and E. E. Hartquist. Digital filtering via block recursion. *IEEE Transactions on Audio Electroacoustics*, AU-18:169–176, June 1970.

[182] S. Waser and M. J. Flynn. *Introduction to Arithmetic for Digital Systems Designers.* Holt, Rinehart and Winston, New York, 1982.

[183] B. Widrow. A study of rough amplitude quantization by means of nyquist sampling theory. *IRE Transactions on Circuit Theory*, CT-3:266–276, December 1956.

[184] D. Williamson. Roundoff noise minimization and pole-zero sensitivity in fixed-point digital filters using residue feedback. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34(5):1210–1220, October 1986.

[185] D. Williamson. *Digital control and implementation : finite wordlength considerations.* Prentice Hall International Series in Systems and Control Engineering. Prentice Hall International, Englewood Cliffs, N.J., 1991.

[186] D. Williamson and K. Kadiman. Optimal finite wordlength linear quadratic regulation. *IEEE Transactions on Automatic Control*, 34(12):1218–1228, December 1989.

[187] D. Williamson and S. Sridharan. Residue feedback in digital filters using fractional feedback coefficients. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33(2):477–483, April 1985.

[188] D. Williamson and S. Sridharan. Error feedback in a class of orthogonal polynomial digital filter structures. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34(4):1013–1016, August 1986.

[189] P. W. Wong. Quantization noise, fixed-point multiplicative roundoff noise, and dithering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(2):286–300, February 1990.

[190] P. W. Wong. Quantization and roundoff noises in fixed-point FIR digital filters. *IEEE Transactions on Signal Processing*, 39(7):1552–1563, July 1991.

[191] C. Xiao. Improved $\mathcal{L}_2$-sensitivity for state-space digital system. *IEEE Transactions on Signal Processing*, 45(4):837–840, April 1997.

[192] D. J. Xu and M. L. Daley. Design of optimal digital filter using a parallel genetic algorithm. *IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing*, 42(10):673–675, October 1995.

[193] G.-T. Yan. New digital notch filter structures with low coefficient sensitivity. *IEEE Transactions on Circuits and Systems*, CAS-31(9):825–828, September 1984.

[194] W.-Y. Yan and J. Moore. On l_2-sensitivity minimization of linear state-space systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(8):641–648, August 1992.

[195] J. Zeman and A. G. Lindgren. Fast digital filters with low roundoff noise. *IEEE Transactions on Circuits and Systems*, CAS-28:716–723, July 1981.

[196] G. Zhu, K. Grigoriadis, and R. E. Skelton. Optimal finite wordlength digital control with skewed sampling. In *Proceedings of the American Control Conference*, volume 3, pages 3482–3486. IEEE, 1994.