A Jump Linear Framework for Estimation and Robust Communication with Markovian Source and Channel Dynamics

by

Alyson Kerry Fletcher

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in

Engineering—Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Kannan Ramchandran, Chair
Professor Venkatachalam Anantharam
Professor Alexandre J. Chorin

Fall 2005

A Jump Linear Framework for Estimation and Robust Communication with Markovian Source and Channel Dynamics

Abstract

A Jump Linear Framework for Estimation and Robust Communication with Markovian Source and Channel Dynamics

by

Alyson Kerry Fletcher

Doctor of Philosophy in Engineering—Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Kannan Ramchandran, Chair

The estimation and reliable transmission of signals in the presence of data losses is a fundamental problem of communication and signal processing. Data losses arise in numerous areas, including wireless communications, communication over lossy packet networks, and estimation problems with intermittent or random sampling.

The traditional approach to mitigate the effect of losses is channel coding. However, for time-sensitive applications, channel coding may not be feasible due to the delay introduced in averaging out the variations in the channel. In place of channel coding, time correlations in signals may inherently provide some robustness to data losses. For example, if a signal is slowly varying, lost samples can be estimated by interpolation. However, for communication and compression problems, correlations also create redundancy in the transmitted data, which may result in an excessive data rate. Basic questions then are: What is the precise value of signal correlations for robustness to data losses? And, how does one trade off robustness and compression?

This work proposes a general framework based on jump linear systems for analyzing the estimation and communication of correlated signals. A jump linear system is a linear state space system whose time variations are driven by a discrete, finite Markov chain. In any single Markov chain state, there are linear, continuous-valued dynamics that describe the signal, as well as any filters in the encoder and decoder and the linear channel. The Markov chain models discrete changes in any part of the system, notably changes in the channel or observation process such as sample erasures. The model is extremely general and can capture arbitrary second-order signal statistics, linear encoders and decoders, and arbitrary Markovian loss processes.

We first consider a general estimation problem for any signal described by a jump linear system. It is shown that the average estimation error can by bounded by a convex programming method based on linear matrix inequalities (LMIs). The LMI result is the mathematical dual to earlier results in jump linear state feedback control.

The LMI optimization provides a simple method for computing an upper bound on the achievable distortion at the estimator as a function of the signal dynamics, encoder, quantization error, and data losses. A simple jump linear estimator that achieves this bound is also presented. This characterization may greatly simplify system design, as prior to this work the only evident means for performance evaluation was Monte Carlo simulation.

The result is then applied to predictive quantization, which is a widely used technique for coding correlated signals. While predictive quantization can improve the compression of a signal, the transmitted data is significantly less robust to losses. Using the LMI method, we can quantify the effect of losses in predictive quantization for a general Markovian loss process and predictive quantization encoder of arbitrary order. A modified predictive quantization encoder is proposed to better trade off robustness and prediction gain.

A simple multiple description (MD) coding method is also proposed, where the robust predictive quantization outputs are separated into odd- and even-numbered samples and transmitted as two independent streams. By varying a robustness parameter of the encoder, one can trade off the distortions between the cases when both streams arrive and when only one stream arrives. The predictive MD coding method is demonstrated by implementing a block-adaptive audio coder.

Professor Kannan Ramchandran
Thesis Committee Chair

To the Simones & Lola

Contents

Contents **ii**

List of Figures **vi**

Acknowledgments **viii**

1 Introduction **1**

1.1 Estimation and Robust Communication of Correlated Signals 1

1.1.1 Basic Questions 1

1.1.2 Channel Coding and Delay 3

1.2 Jump Linear Framework 5

1.2.1 Linear Markov Model 5

1.2.2 State Space Representation 7

1.3 Main Results 9

1.3.1 Jump Linear Estimation 9

1.3.2 Robust Predictive Quantization 12

1.3.3 Multiple Description Predictive Quantization 14

1.4 Outline 15

1.5 Notation 15

2 Background: Linear Systems, Estimation, Optimization and Quantization **17**

2.1 State Space Models for Random Processes 17

2.2 Stability, Stabilizability and Detectability 20

2.3 Lyapunov Equation 21

2.4 Kalman State Estimation 23

2.4.1 Basic Equations 23

2.4.2 Systems with Known Inputs 25
2.4.3 Algebraic Riccati Equation and the Steady-State Filter 26
2.4.4 One-Step Ahead Prediction 28
2.4.5 Noncausal Estimation 29
2.5 Denoising with Kalman Filtering 31
2.6 Linear Matrix Inequalities (LMIs) 32
2.7 LMI Approach to Linear Estimation 34
2.7.1 Lyapunov Analysis 34
2.7.2 Strictly Causal Kalman Filtering 36
2.7.3 Causal Kalman Filtering 39
2.8 Basic Principles of Quantization 42
2.8.1 Definitions and Performance of Simple Fixed-Rate Quantizers 42
2.8.2 Entropy-Coded Quantization 44
2.8.3 Lloyd–Max Quantization 45

3 Estimation for Jump Linear Systems **47**
3.1 Introduction 47
3.1.1 Jump Linear Estimation 47
3.1.2 Summary of the Main Results 49
3.1.3 Previous Work 50
3.2 Mean Square Stability and Detectability 53
3.3 Coupled Lyapunov Equations 54
3.4 Lyapunov Analysis 56
3.5 Jump Linear Estimation 59
3.6 LMI Analysis 60
3.6.1 Strictly Causal Estimation 60
3.6.2 Causal Estimation 63
3.7 Coupled Riccati Equations 64
3.8 Independent Markov Chains 66
3.9 Proofs 71
3.9.1 Proofs of Theorems 3.1 and 3.2 71
3.9.2 Proof of Theorem 3.3 73

4 Estimation with Markovian Losses **78**

4.1 Introduction 78
4.1.1 Related Work in Sensor Networks 79
4.2 Jump Linear Modeling 81
4.3 LMI Analysis 83
4.4 Independent Losses 85
4.5 Algebraic Riccati Equation for I.I.D. Losses 88
4.6 Extensions 89
4.7 Numerical Examples 90
4.7.1 Denoising with Lossy Observations 90
4.7.2 Two-Dimensional Tracking 91

5 State Space Design for Predictive Quantization 95

5.1 Predictive Quantization 97
5.1.1 Autoregressive Sources and Prediction 98
5.1.2 History and Prior Analyses 99
5.2 Additive White Noise Quantizer Model 101
5.3 Linear State Space Predictive Quantizer System Model 104
5.4 Encoder and Decoder Filter Design Problem 105
5.5 Kalman Filter Solution 107
5.6 High Rate Limit and One-Step Ahead Prediction 111
5.7 Numerical Example 112
5.8 Relation to Rate–Distortion and Optimal Sample Rate 114
5.8.1 High Rate Comparison 114
5.8.2 General Rate Analysis 117

6 Robust Predictive Quantization 120

6.1 Prediction vs. Robustness: First-Order Example 121
6.2 State space Model 123
6.3 Jump Linear Decoding and Analysis 125
6.3.1 Encoder Gain Matrix Feasibility 125
6.3.2 Jump Linear Decoder 127
6.3.3 LMI analysis 130
6.4 Encoder Gain Optimization 131
6.5 Numerical Example 133

7 Multiple Description Coding by Sample Separation **136**
7.1 Introduction to Multiple Description Coding 137
7.1.1 History 140
7.1.2 Comparison to Robust Predictive Quantization 141
7.2 The Odd-Even Separation Method 142
7.3 Analysis and Design with LMIs 144
7.3.1 Odd-Even Separating MD Encoder 144
7.3.2 Jump Linear Analysis 145
7.3.3 Encoder Gain Optimization 148
7.4 Numerical Example 149
7.5 MD Audio Coding 150
7.5.1 AR Models and the Yule-Walker Equations 151
7.5.2 Standard AR Encoder and Decoder 152
7.5.3 Proposed MD Audio Predictive Coder 154
7.5.4 Audio Example 157
7.6 Sample Rate Optimization 160

8 Conclusions **162**
8.1 Summary 162
8.2 Future Extensions 163
8.2.1 Estimation theory 163
8.2.2 Robust Predictive Quantization and MD Coding 164
8.2.3 Audio and Video Applications 165
8.3 Other Connections 165
8.3.1 Uncoded Communication 165
8.3.2 Compression of Unstable Processes 166
8.3.3 Wyner–Ziv Coding 166
8.3.4 Applications to Finite Bit Rate Control 167

Bibliography **167**

List of Figures

1.1 Communication of a correlated signal with simple scalar quantization and a lossy channel. 2

1.2 Communication with predictive quantization. 3

1.3 Communication using the separation principle. 4

1.4 Linear model of communication with a Markov-varying channel. 6

2.1 Reconstruction SNR of a Gaussian source with various quantizers. 44

4.1 Lossy data estimation problem. 78

4.2 Jump linear model for lossy data estimation problem. 81

4.3 Reconstruction SNR of a lowpass signal from noisy samples with i.i.d. losses. 91

4.4 Reconstruction SNR of a lowpass signal from noisy samples with fixed-length losses. 92

4.5 Reconstruction SNR of a two-dimensional lowpass signal with lossy random one-dimensional projections. 93

5.1 Differential quantizer encoder and decoder. 97

5.2 Predictive quantizer encoder and decoder with a general higher-order filter. 98

5.3 Linear model for the predictive quantization system. 105

5.4 Estimation of predictive coding performance with linear AWN model and uniform quantization. 113

5.5 Estimation of predictive coding performance with linear AWN model and entropy-coded scalar quantization. 114

5.6 Predictive quantization with downsampling. 117

6.1 First-order predictive quantizer encoder and decoder with a lossy channel. 121

6.2 General predictive quantizer encoder and decoder with a lossy channel. . . . 123

6.3 Quantization of a lowpass signal with i.i.d. channel losses. 133

6.4 Quantization of a lowpass Gaussian signal with Gilbert-Elliot channel losses. 135

7.1 Multiple description coding with two descriptions. 138

7.2 Best possible side distortion as a function of redundancy for balanced MD coding. 139

7.3 MD coding by odd-even separation as proposed by Jayant. 143

7.4 MD coding by odd-even separation as proposed by Ingle and Vaishampayan. 144

7.5 LMI design of odd-even sample separation MD coder. 145

7.6 MD coding of a lowpass Gaussian signal. 149

7.7 MD coding of an audio signal. 158

Acknowledgments

There are a number of people without whom this dissertation might not have been written and to whom I am greatly indebted.

This work could not have been possible without an advisor who encouraged me to explore questions initially outside the scope of his interest and then lent his boundless energy and creative intuition. For this freedom and his unwavering support through many ordeals, I thank Professor Kannan Ramchandran, my advisor.

This research was improved through my many interactions with Dr. Sundeep Rangan and Professor Vivek Goyal, both inexhaustible sources of ideas and insight, though I fear I lacked the dedication and energy to have made full use of their talents. I thank them for their time, feedback, and wisdom, but even more importantly for making research a joy. I would also like to thank my dissertation and qualifying exam committee members—Professors Alexandre Chorin, Venkat Anantharam, Laurent El Ghaoui, and Kameshwar Poolla—for providing their valuable input. I am also grateful to Professors Andy Packard and Anant Sahai for their time, helpful technical comments, and positive words.

Ruth Gjerde, Mary Byrnes, and Erin Reiche were thoughtful and absolutely essential in navigating the choppy waters of the UCB bureaucracy. Ruth's unfailing dedication to helping students exceeds all rational expectations. Also, Eleta Cook and Loretta Lutcher adeptly and cheerfully helped me with my Cory Hall problems, even those of my own making. I am so appreciative of not just the help, but also the great conversations.

I am forever indebted to Adriana Schoenberg, Dr. Jeff Nelson, Dr. Gerald Keane, and most importantly, the incomparable John King, for trying to keep me healthy enough not just to complete this dissertation, but to live outside the work. This was not an easy task, for me, or for them.

I gratefully acknowledge Professor Martin Vetterli for his encouragement and his ability to make me feel that I mattered when I most needed it. Professor Serap Savari, Dr. Emina Šoljanin, and especially, Professor Jelena Kovačević were amazing and encouraging "big sisters" at every juncture. Professors Nick Kingsbury and Amin Shokrollahi showed an interest in my research and allowed me the opportunity to travel to wonderful places and present my work.

I was lucky to enjoy the presence of a supportive research group in both BASiCS and Wireless Foundations. I couldn't have done this without a few friends who tried to both keep me sane and make me have some fun: Mareike Claassen, Leon Abrams, James Yeh, DeLynn Bettencourt, June Wang, Dave Nguyen, Kristie Korneluk, Anand Sarwate, Stark Draper, Prakash Ishwar, Andreas Anagnostopoulos, Dan Hazen, Aaron Wagner, Ken Chiang, Michael Gastpar, Adam Frank, Gabe Moy, and the extraordinary Moodle. I have been so lucky to have Mark Johnson around, not just for his helpful spirit, but for his unfailingly perfect dry humour, without which the days would have been far bleaker.

This path began long ago in a seemingly distant land, and there are too many people who mattered to include them all. I would not be who I am without my sisters—Lorraine, Leslie, and Heather—and Wendi, my companions in exile and partners in the imagination that started me on my path. My parents instilled in me a love for reading and the belief in and desire for a different life. Also in memoriam to Professors Christoph Goertz and Kenneth Sando, for their efforts on my behalf; the academic world should have more such as them. I am indebted to Professors William Klink, Kevin Campbell, Paul Kleiber, Milan Sonka, Soura Dasgupta, and Kendall Atkinson for their teachings and the opportunities they gave me. The camaraderie of my gang—Tim Shoppa, Jackson Loomis, and Tim Olander—was essential. Hassime Traore was a smiling face and my guide through Quantum Chemistry. It wouldn't have been the same without Joe Lazio, an irreplaceable soul, who was a good friend and a weird roommate, and Jennifer Wuggazer—a friend and Joe's other woman. Everyone else in LASA, AIHS, Shambaugh Honors House, the Campbell Lab, and Van Allen Hall knows who they are.

Thank you to my second family, Ron & Sally Goldstein, and Anna, Jane & Alan Schoenfeld for loving me even though they don't have to and giving me places to call home.

Lola made getting out of bed every morning and coming home at night sheer joy. That we found each other remains one of life's happiest surprises. Finally, that this dissertation was finished at all is due in large part to Sundeep and Vivek in their other more important role, as my dearest friends. The help, encouragement, support, and care that Sundeep gives me every day is astounding and more than one could ever ask for or expect in a lifetime. Without the laughter and kindness that he brings to both my worst and best days, my world would not be the same. But most of all, I not only couldn't have written this dissertation but I wouldn't even be here at all without the unwavering belief and support of Vivek, my oldest and closest friend and a source of encouragement and inspiration throughout my life. He is the most unconditionally generous person on earth. My life is so much the richer for knowing both of them.

My deepest thanks to all.

Chapter 1

Introduction

1.1 Estimation and Robust Communication of Correlated Signals

1.1.1 Basic Questions

The estimation and reliable transmission of signals in the presence of data losses is a fundamental problem of communication and signal processing. In communication, losses can arise from physical variations in the channel, such as Rayleigh fading in wireless links [126], or packet losses due to network congestion and loading fluctuations. In signal processing, data may also be incomplete due to intermittent or random sampling.

The purpose of this dissertation is to study the estimation and robust communication of signals with time correlations. Time correlations result in signal memory, which may inherently provide some degree of robustness to data losses. As an example, consider first the simple communication system in Figure 1.1. Here $z[k]$ represents a correlated random process whose values are to be communicated over a lossy channel. The encoder simply quantizes each sample of $z[k]$ and thereby generates a sequence of digital samples, $q[k] = Q(z[k])$, that are transmitted over the channel. In place of the lossy channel, one could also consider a random sampling of the signal.

Now, suppose the channel has losses, so that some of the samples $q[k]$ are erased and not received at the decoder. Since the process $z[k]$ is correlated, the signal itself provides

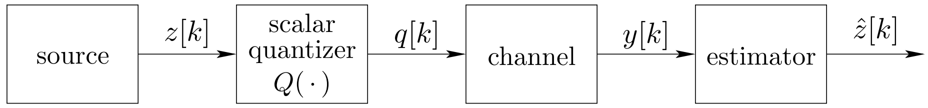

Figure 1.1. Communication of a correlated signal with simple scalar quantization and a lossy channel.

some degree of robustness. For example, if the signal is slowly varying, then neighboring samples of $z[k]$ should be close in value. Consequently, the decoder may be able to estimate the value of lost samples by interpolating from neighboring samples that were received.

However, correlations in the signal $z[k]$ result in the information in the transmitted data $q[k]$ being *redundant*. That is, the information in any sample $q[k]$ can be at least partially estimated from other samples. While this redundancy provides robustness to losses, the redundancy also implies that the samples $q[k]$ are not an efficient representation of the signal. Theoretically, by removing the redundancy one should be able to improve the compression of the source, and thereby either improve the fidelity of the representation or reduce the data rate.

One way to reduce the redundancy of the samples $q[k]$ (improve the compression) is through *predictive quantization* [64]. In predictive quantization, the scalar quantization is performed in feedback with a linear filter, as shown in Figure 1.2. The encoder filter $H_{\rm enc}$ attempts to "subtract out" correlations between current samples $q[k]$ and previous samples to improve the overall compression. For example, in the case of a slowly-varying signal, predictive quantization could be used to quantize only the difference between the signal and the previous quantized value, as opposed to quantizing the samples themselves. If the signal is highly correlated, then predictive quantization can significantly improve the efficiency of the representation. Predictive quantization is a widely-used technique for signal compression, particularly for speech and video.

Unfortunately, predictively quantizing data makes it significantly more susceptible to losses. Due to the feedback filter $H_{\rm enc}$, the output samples $q[k]$ implicitly depend on past samples. Consequently, loss of any one sample tends to affect future samples and degrade the overall performance of the coding system significantly. More generally, one could say that predictive quantization, by design, attempts to remove the redundancy in the quantized data, thereby losing the inherent robustness of the signal to losses.

We thus see that in the transmission of a correlated source, there is a fundamental trade-off between leaving the redundancy for robustness, and removing the redundancy

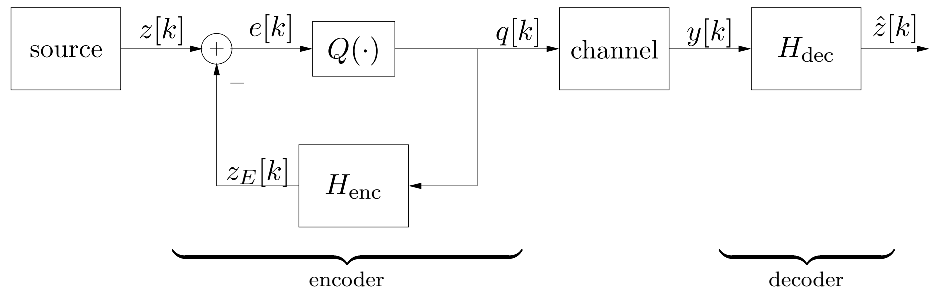

Figure 1.2. Communication with predictive quantization.

to achieve better compression. The scalar quantization in Figure 1.1 that leaves all the signal redundancy and the predictive quantization in Figure 1.2 that subtracts out all the redundancy represent two extremes. Clearly, there is some middle ground.

In this context, the goals of this dissertation are twofold:

1. To understand the value of signal memory in mitigating data losses, allowing for rich signal and channel dynamics. Specifically, we wish to develop methods for evaluating the effect of data losses in signal estimation, as a function of the loss and signal statistics as well as other transforms, filtering, and quantization that may be present in the signal encoding or observation.

2. For communication problems, to precisely quantify the trade-off between compression and robustness, and to develop better methods to achieve this trade-off while exploiting the inherent redundancy in the signal.

At the core of the contributions of the dissertation is the modeling of communication and estimation problems via jump linear state space systems. New results on optimal estimation in such systems—specifically, constructive estimation error bounds that are easy to compute using convex optimizations with linear matrix inequality constraints—are first developed. Then, these results are used to analyze and optimize robust communication systems.

1.1.2 Channel Coding and Delay

In communication problems, the conventional approach to dealing with losses is to protect the quantized data with *channel coding.* The use of channel coding in conjunction with source coding is illustrated in Figure 1.3. The source code represents blocks of the

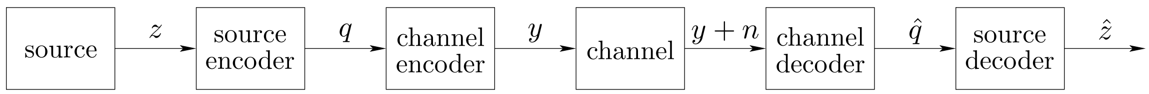

Figure 1.3. A communication system using the separation principle. Consider z and $\hat{z}$ to be in a continuous-valued space equipped with a fidelity measure. The representation q is in bits and should be reconstructed exactly (at least with overwhelmingly high probability) as $\hat{q}$. The channel adds noise n to the channel input y; note in particular that n could depend on y, for example cancel y.

source sequence with blocks of bits, and the channel code maps blocks of bits to channel inputs. The source code attempts to represent the data with a minimal number of bits, and the channel code attempts to maps those bits to the channel input to guarantee a high probability of reliable reception.

One of the classical and fundamental results of information theory is Shannon's *separation principle:* under very general conditions, the separation of the communication problem into efficient representation of the information as bits (source coding) and reliable conveyance of bits (channel coding) is theoretically optimal and incurs no loss in performance with respect to any more general joint source and channel coding [135].

Using the separation principle has several important features. First and foremost, the separation principle decouples the problems of compression and robustness. Moreover, it establishes bits as a universal currency for point-to-point communication, without regard for the source, channel, or fidelity criterion. In particular, it is not necessary for the number of source symbols per second to equal the number of channel uses per second. As a practical matter, separation is synonymous with modularization, with separate design and easy reuse of the component source and channel codes.

The main disadvantage of separation is that good performance generally requires a channel code that works on long blocks and hence contributes to a large communication delay. Source coding can also introduce delay, although this can be kept small without calling into question the separation paradigm. In particular, predictive scalar quantization does not contribute to delay. Delay in channel coding is more fundamental since, in order that channel coding guarantees reliability, the coding blocks must be sufficiently long to average out the variations of the channel. Along with the delay, channel coding with long block-length codes can increase the decoder complexity.

Consequently, in delay sensitive applications, channel coding may not be feasible.[1] In

[1]It is possible to extend the results of this dissertation to situations in which channel codes are used, but

such scenarios with delay constraints, the separation principle no longer holds, and the problems of the compression of the source and robustness to channel impairments become tightly coupled. Thus, with strict delay requirements, communication systems must often employ *joint* source and channel coding.

The need for joint source and channel coding for low delay is well known, and there is an enormous body of work in this area. The analysis in this dissertation is closest to [6, 17, 24, 80, 131], which consider discrete-valued sources sent over discrete memoryless channels. These works exploit the source redundancy for robustness to channel losses to achieve low-delay reliable transmission.

This dissertation approaches the joint source channel coding problem in a similar vein, but with continuous-valued correlated signals communicated over channels with more general Markovian dynamics. The focus here, however, is less on the information theoretic aspects of this problem. Instead, our goal is to develop practical methods for reliable communication with minimal delay, employing channel and signal models that are both general and widely used. The dissertation offers a new analysis and design framework; even where the encoder/decoder design methodology for particular links is not employed, the analyses that are newly made tractable may be important in system design.

1.2 Jump Linear Framework

1.2.1 Linear Markov Model

Given the complexities of the coupling of signal and channel dynamics and the nonlinearity of source coding, it is exceedingly difficult to derive a general theory for estimation and robust communication over lossy channels. What is needed is a model that is sufficiently general to study interesting source and channel dynamics, but also amenable to quantitative analysis and optimization. To this end, we propose a model that combines two classical ideas in signal processing: linear signal models and discrete Markov chains. Taken together, this linear Markov model affords a great deal of generality, while simultaneously being based on components that have been analyzed extensively and are ubiquitous in both engineering theory and practice. This will allow us to derive concrete quantitative

they are too short to make the probability of decoding failure negligible. This is not considered explicitly, as the more extreme case of having no channel code is more straightforward to discuss.

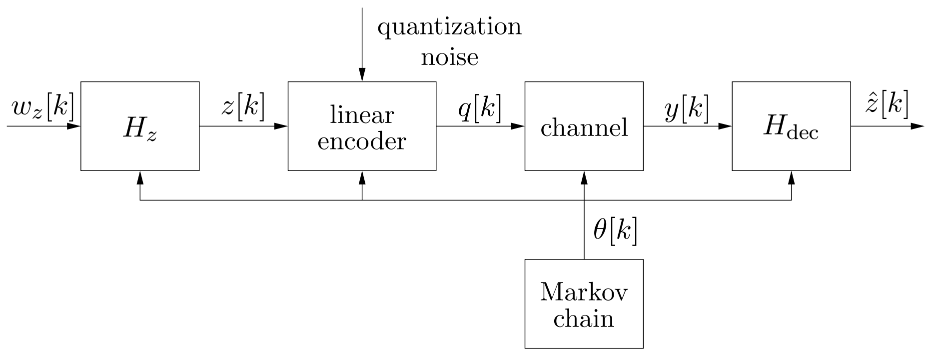

Figure 1.4. General communication system model with linear signal dynamics, additive quantization noise and a Markov-varying source and channel dynamics.

results, while showing how to improve upon standard signal processing and communication methods.

The general model we consider is shown in Figure 1.4 and consists of the following components:

- The signal $z[k]$ is modeled as filtered white noise: $z = H_z w_z$. This is a standard model in linear signal processing and is well-suited to wide-sense stationary random processes characterized by their second-order statistics. The model is general in the sense that the filter $H_z(z)$ can be selected to capture an arbitrary power spectral density of $z[k]$. For example, if $z[k]$ is lowpass, then $H_z(z)$ can be selected as a lowpass filter whose cutoff frequency equals the signal bandwidth.

  In our source model, the filter $H_z(z)$ may depend on the Markov chain $\theta[k]$. When $\theta[k]$ is constant, the filter is time-invariant; the possibility of dependence on a time-varying $\theta[k]$ enriches the model to include piecewise ARMA signals.

- The quantizer is also modeled as a linear transform, with the effect of the quantization modeled as additive noise. Of course, quantization is inherently a deterministic nonlinear operation, so modeling it as the addition of signal-independent noise is not accurate for all purposes. Nevertheless, such linear additive noise models for quantizers are widely used and have proven to be accurate, at least in terms of capturing the second-order statistics of the quantization error when the quantization is not too coarse [74, 88, 98]. As an analytical tool, one can dictate the use of subtractive dithered quantization in order to make the quantization noise truly independent of the quantizer input [76, 102]. This distinction is not important in practice. The essential limitation is that we use a linear decoder; when using a linear decoder, only the second-order statistics of the quantization noise are needed.

The linear quantizer model can also incorporate any transforms and filters in the quantization, such as the subtractive filtering in linear predictive quantization. Dependence on $\theta[k]$ is again allowed.

- The channel is modeled as a linear channel whose time variations are driven by a finite-state Markov chain, $\theta[k]$. This means that in any given state, the channel is linear with an additive noise. The states may correspond to different signal-to-noise ratios including, for example, zero signal-to-noise ratio for an erasure. The Markov chain model is extremely general and can capture a wide variety of correlated loss processes, such as burst losses.

  One of the motivations for the Markov model is that the channel dynamics are especially important to model correctly when channel codes are not used. With long block-length channel codes, the importance of local variations of the channel is diminished and only the typical behavior is essential. In the limit of infinite delay, channel coding can always achieve the average channel capacity. However, for communication with channel coding, the time dynamics of the channel become critical since the coding cannot buffer the data and must deal with the instantaneous channel capacity instead of the average capacity.

- Given the channel output $y[k]$, the decoder forms an estimate $\hat{z}[k]$ of the signal $z[k]$. Since we have assumed linear signal, quantizer, and channel models, we can restrict the decoder to be linear as well. We also assume that the state $\theta[k]$ is known to the decoder. When $\theta[k]$ represents only the state of an erasure channel, the decoder knows $\theta[k]$ because it knows which samples arrive. This case dominates due to the ubiquity of packet networks.

1.2.2 State Space Representation

Taking the components of Figure 1.4 together, we see that signal model, quantizer and channel can be seen as one large linear, time-varying system driven by a Markov chain $\theta[k]$. Such systems have been studied extensively in control and are often called *jump linear systems*, with the adjective "jump" referring to the fact that changes in the Markov state $\theta[k]$ result in discrete changes, or jumps, in the system dynamics.

Jump linear systems and some of the history of their development will be reviewed in Chapter 3, but it is suffices to say here that the systems have been studied since the 1960s and now there is a considerable body of literature on the subject. A comprehensive survey of the area has been recently published in a book by Costa *et al.* [35]. Jump linear systems have found applications in a number of areas, such as fault detection, process control, guidance, and more recently, finance and economic modeling [4, 10, 14, 45, 143]. The main idea in this dissertation is that jump linear systems provide a powerful and general model for understanding linear communication problems as well.

While linear systems are often analyzed in the frequency domain in terms of their transfer functions, jump linear systems are more conveniently represented in *state space form*, such as:

$$\begin{aligned} x[k+1] &= A_{\theta[k]}x[k] + B_{\theta[k]}w[k], \\ z[k] &= C_{1,\theta[k]}x[k] + D_{1,\theta[k]}w[k], \\ y[k] &= C_{2,\theta[k]}x[k] + D_{2,\theta[k]}w[k]. \end{aligned} \tag{1.1}$$

In this representation, all the signals—$w[k]$, $x[k]$, $y[k]$, and $z[k]$—are potentially vector-valued, and the factors $A_{\theta[k]}$, $B_{\theta[k]}$, etc., are compatibly-dimensioned matrices. Throughout this work, we will assume that the Markov chain $\theta[k]$ takes on one of M values, $\theta[k] \in \{0, \ldots, M-1\}$. Consequently, the jump linear system is defined by M sets of system matrices,

$$(A_i, B_i, C_{i1}, C_{i2}, D_{i1}, D_{i2}) \qquad \text{for} \quad i \in \{0,\ 1,\ \ldots,\ M-1\}.$$

Given a realization of the Markov sequence $\theta[k]$, the system (1.1) defines a linear time-varying mapping from the input $w[k]$ to the outputs $z[k]$ and $y[k]$, with an internal state $x[k]$.

For the communication system in Figure 1.4, the system input $w[k]$ would be a vector containing both the source signal input $w_z[k]$ and the quantization noise $w_q[k]$. The system matrices, A_i, B_i, etc., and the state $x[k]$ model the dynamics in the signal generating filter $H_z(z)$, along with any linear dynamics in the quantizer and channel. The Markov chain $\theta[k]$ captures discrete changes in the channel condition.

Jump linear systems of the form (1.1) have been studied extensively, and the motivation for describing the communication problem as a jump linear system is to leverage this theory for our problem. For example, consider the problem of the decoder design. The decoder must estimate the transmitted signal $z[k]$ from the received signal $y[k]$. In the jump linear system (1.1), this problem is equivalent to estimating one output from another output. This problem is, in fact, the standard jump linear estimation problem and we study it in detail.

Using these results, we can thus understand how one should design the decoder, and what performance one should expect as a function of the signal and loss statistics.

The idea of leveraging jump linear systems for a communications problem is part of a broader goal of this research: to connect ideas in state space analysis specifically, and control theory more generally, to communications problems. Much of classical linear signal processing and communications of correlated random processes is analyzed in the frequency domain. The frequency domain approach is analytically simple and uses intuitive quantities, such as transfer functions and power spectral densities. For these reasons, state space models are rarely used in communications and signal processing problems.

However, one of the major limitations of the frequency-domain approach is that it is most naturally suited to time-invariant systems.[2] Consequently, time variations in the system, such as those arising in a lossy channel, cannot be captured easily in the frequency domain. In contrast, the jump linear state space model (1.1) represents both the linear time-invariant signal dynamics and time-varying channel dynamics in a single time-domain representation. This ability of state space models to capture complex time variations and dynamics may be of value to other communications problems as well, and it is hoped that the research in this dissertation will encourage this study.

1.3 Main Results

1.3.1 Jump Linear Estimation

For the communication system in Figure 1.4, the basic problem is how to design the encoder, given the model for the source and channel. To consider this problem, it is desirable to have a method to compare various candidate designs. A natural metric for this comparison is: given models for the source and channel, and a candidate encoder, what is the minimum distortion achievable at the decoder? If we have a simple method for evaluating this metric, we can compare various encoder designs and select the one that minimizes the average distortion.

The most common distortion measure—and the one that we will use throughout this work—is the mean-squared error (MSE) between the transmitted signal $z[k]$ and decoder

[2] Frequency domain analysis can be extended from time-invariant systems to periodically time-invariant systems through the use of a polyphase representation.

output $\hat{z}[k]$:

$$\text{MSE} = \mathbf{E}\left(\|z[k] - \hat{z}[k]\|^2\right). \tag{1.2}$$

Here the expectation is taken over both realizations of the random signal $z[k]$ and realizations of the channel.

Now, the decoder must compute the estimate $\hat{z}[k]$ given the channel output $y[k]$ and channel state $\theta[k]$. Assuming that we restrict our attention to causal estimators, the optimal estimator at the decoder for the error criterion (1.2)—the minimum mean-squared error (MMSE) estimator—is

$$\hat{z}[k] = \mathbf{E}\left(z[k] \mid y[0:k], \theta[0:k]\right), \tag{1.3}$$

the conditional expectation given the channel output $y[j]$ and channel Markov state $\theta[j]$, for time j from 0 to k.[3] Since the estimator knows $\theta[k]$, the estimator essentially sees a linear system with known time variations. Moreover, the estimate (1.3) is precisely the MMSE estimate of one of the system outputs, $z[k]$, from a second output $y[k]$. This is a standard estimation problem whose solution is given by the well-known *Kalman filter*.

The Kalman filter was originally developed in the 1960s in [90, 91] and is covered in several textbooks such as [9] and [97]. Given a jump linear system (1.1), the Kalman filter provides simple, recursive equations for the estimate $\hat{z}[k]$ along with the output error variance

$$\sigma^2[k] = \mathbf{E}\left(\|z[k] - \hat{z}[k]\|^2 \mid \theta[0:k]\right). \tag{1.4}$$

The equations will be reviewed in Section 2.4.

The Kalman filter solution would appear to be a complete solution to the problem. However, the variance $\sigma^2[k]$ in (1.4) is the error variance sequence for a *given realization* of the Markov chain $\theta[k]$. In the context of the communication problem, this would provide the error variance for a particular realization of the channel. To evaluating a proposed communication scheme, we need the *average* error variance, averaged over the realizations of $\theta[k]$.

Computing this average error as a function of the Markov chain statistics is, in general, difficult since the update equations for the error are highly nonlinear. Our main results, Theorems 3.1 and 3.2, provide upper bounds for computing the average error via a convex optimization procedure based on linear matrix inequalities (LMIs). Theorem 3.1 considers

[3] We use a MATLAB-inspired notation of $y[0:k]$ to denote $(y[0], y[1], \ldots, y[k])$. Other notational conventions are described in Section 1.5.

the case of strictly causal estimation, and Theorem 3.2 considers causal estimation. A simple jump linear estimator that achieves these bound is also provided.

LMI optimization can be seen as a matrix-valued generalization of linear programming. It has been widely used in control theory since the 1990s. The book by Boyd *et al.* [16] provides a comprehensive overview of the subject. The use of LMIs in jump linear analysis is not new and dates back to the work of Ait Rami and El Ghaoui [3]. Also, Costa *et al.* [31] consider jump linear state feedback control, which is mathematically the "dual" to state estimation. Our derivation, similar to [31], is based on the Lyapunov analysis in [34]. The final LMIs we obtain for the estimation problem are a natural dual to the control LMIs in [31].

Our main results are general and apply to an arbitrary jump linear system driven by a stationary Markov chain. Therefore, the results may have applications beyond communications problems and may be useful wherever jump linear systems are applied. A general review of jump linear systems and proofs of our results are given in Chapter 3.

The results are then applied to evaluating the effect of losses in estimation in Chapter 4. We show that the jump linear model is extremely general and can incorporate correlated loss processes, variable channel qualities, signal dynamics, and multiple observations. Numerical examples also show that, for certain models, our upper bounds on the estimation error are very accurate performance estimates.

Computing the effect of losses on the average estimation error has also been recently studied by Sinopoli *et al.* [139, 140] and Liu and Goldsmith [103]. These works arose in the study of sensor networks. The results consider state estimation where the observations are subject to i.i.d. erasures. It is shown, surprisingly, that if the underlying system is unstable, there is a critical loss probability above which the state cannot be tracked with finite variance. A simple upper bound on this loss probability is given in terms of the maximum eigenvalue of the system A matrix.

An LMI is also provided whose feasibility gives a sufficient condition for stability and an upper bound on the average estimation error. Sinopoli *et al.* in [139, 140] consider a single lossy observation, and Liu and Goldsmith in [103] extend the LMI to the case to two observations. We will show in Chapter 4 that both results are special cases of the more general jump linear estimation LMI we present, which can consider arbitrary Markov loss processes and multiple observations.

1.3.2 Robust Predictive Quantization

A simple yet extremely effective technique for compressing slowly-varying signals is predictive quantization, in which prediction errors are coded rather than the raw samples. Since the prediction errors have lower variance than the raw samples themselves, the quantization error is reduced. If the signal is slowly varying, then predictive quantization can realize significant gains.

Predictive quantization is the workhorse of source coding, forming the basis of most speech and video coding. It is used with adaptation of the quantizer step size for compression of speech signals at rates from 16 kbits/sec to 64 kbits/sec [65], and it is the basis of motion-compensated video coding, as in MPEG [100], H.264 [128], and other standards. Furthermore, when delay or latency is a key consideration, predictive quantization is the best technique for audio compression [132].

At an information theoretic level, the gains in compression from predictive quantization come from subtracting out the redundancy in a correlated signal. However, as discussed in Section 1.1, this loss in redundancy also makes predictively quantized data less robust to losses when the quantized data is transmitted without the protection of channel coding. The transmission of predictive quantization data over lossy channels is thus a natural place to start in understanding the trade-off between robustness and compression of correlated signals. One of the main results in this dissertation is that the jump linear estimation analysis described above can be used to quantify the effect of losses in predictive quantization, and to redesign the predictive quantization system for improved robustness.

Our analysis begins in Chapter 5, where we revisit standard predictive quantization without losses, but from a state space perspective. Of course, predictive quantization has been studied extensively since the 1970s and is covered in early tutorials such as [85, 108], as well as the standard textbook on quantization [64]. However, the standard approach to studying predictive quantization is in the frequency domain. This work develops a state space theory of predictive quantization that is novel and better suited to generalizing to lossy channels. For the lossless case, the state space results are, however, largely not new *per se*, but rather restatements of well-known facts in state space form. For example, we rederive the well-known separation of prediction gain and coding gain in [64], as well as the asymptotic optimality of predictive quantization at high rates in terms of the rate–distortion of a correlated signal.

A minor benefit of the state space approach for the lossless analysis is that we are able to provide simple state space expressions for the optimal encoder and decoder filters that incorporate the effects of quantization in the feedback loop. In the frequency domain, the predictive filter in the encoder is typically designed ignoring the feedback quantization noise [5]. The closed-loop optimization is difficult, although somewhat complicated methods are available in [93] and more recently, [78]. We show that closed-loop optimization is easy in state space. Moreover, the closed-loop state space expressions reveal an interesting connection: the effect of quantization is, in a certain precise sense, identical to the effect of losses with no quantization.

Combining the state space modeling of predictive quantization in Chapter 5 with the LMI jump linear analysis of lossy channels in Chapters 3 and 4, we can precisely quantify the effects of losses in predictive quantization. Due to the generality of the jump linear model, we can incorporate any Markovian loss process, and arbitrary linear signal models and prediction filters. This provides the best technique to date for quantifying the effect of losses in predictive quantization.

As a minor limitation in the analysis, it should be pointed out that the approach is fundamentally based on an additive white noise model approximation for the quantizer. This approximation is not theoretically exact: indeed, it is well-known that the spectral characteristics and distribution of the quantization noise may not be white. However, the exact distribution is complex and difficult to compute [8, 56], and the approximation we use is standard and widely used. Practical experience, including our simulations, show that the white noise approximation is extremely accurate. The real limitation is that the decoder is restricted to be linear.

The LMI analysis quantifies the performance of predictive quantization for a *given* encoder. To improve the robustness of predictive quantization, we propose a simple modification to the standard predictive quantization encoder filter that reduces the prediction gain but leaves some of the redundancy in the signal for robustness. The proposed robust predictive quantizer is characterized by a single parameter $\lambda \in [0, 1]$ that can be adjusted to trade off robustness and compression. A simple simulation shows that the robust predictive quantization can offer significant improvements to standard predictive quantization in the presence of losses. The analysis of predictive quantization with losses and the proposed robust predictive encoder are described in Chapter 6.

1.3.3 Multiple Description Predictive Quantization

An interesting application of the proposed robust predictive quantization encoder is for a robust source coding technique known as *multiple description* (MD) coding. MD coding is a method for robustly coding data over lossy packet networks. The source data is coded into a number of data streams, called *descriptions*, with each description being transmitted separately. Communication losses may result in one or more of the descriptions failing to arrive at the decoder. In MD coding, the descriptions are designed for graceful degradation with losses: losses in the descriptions will result in a degradation in the received quality, but not a complete failure. Thus, receivers will see a variable quality depending on the quality of the communication link. A survey of MD coding can be found in [69].

In Chapter 7, we apply the robust predictive quantization design to a paradigm for MD coding original proposed by Jayant [86] for speech and applied more recently for video [152, 82]. In the coder in [86], the speech data is first predictively quantized, and the quantized samples are then divided in two descriptions: one consisting of the odd samples, and the other of the even samples. This creates two descriptions, each at half the rate of the original predictive quantizer output. If one of the streams is lost, the decoder can estimate the missing samples from interpolation of the samples that did arrive. Moreover, by adjusting the predictive filter gain, Jayant shows that one can trade off the distortion when both stream arrive (the central distortion), and the distortion when only one stream arrives (the side distortion).

All published analyses of Jayant's scheme are limited to first-order autoregressive sources. Even for this simple source, the analysis is extremely complex. However, from the perspective of the jump linear analysis of predictive quantization, dropping of every other sample is just one type of Markovian loss process. We can thus apply the robust predictive quantizer to obtain an MD encoder that applies to higher-order signal models. The robustness parameter λ in the robust predictive quantizer trades off the central and side distortion and has a similar role as Jayant's scalar prediction gain.

The proposed robust predictive MD coder is demonstrated on both a synthetic Gaussian source and a standard audio test signal.

1.4 Outline

The outline of this dissertation is as follows. To make the dissertation relatively self-contained, we begin in Chapter 2 with a review of some basic facts from linear systems theory, Kalman estimation, LMI optimization, and quantization. Almost all the results of this chapter are well known, and references are provided for the interested reader.

Chapter 3 covers the basic material for jump linear systems and derives our main jump linear estimation results, Theorems 3.1 and 3.2. The results are then applied to quantify the effect of losses in signal estimation in Chapter 4.

Chapter 5 provides a review of predictive quantization, but from a state space perspective. Most of the results are not new *per se*, but restatements of classical results in state space form. As minor new results, we provide a simple state space formula for computing the effect of closed-loop quantization noise. The formula shows an interesting relationship between quantization noise and losses.

Chapter 6 combines the jump linear estimation theory in Chapter 3 and the state space predictive quantization analysis in Chapter 5 to quantify the effect of losses in predictive quantization. This is, perhaps, the most novel and significant contribution in the dissertation. A simple method for trading off the robustness to losses and prediction is also provided.

Chapter 7 extends the robust predictive quantization design for a sample-separating MD coder. The design is tested on a standard audio file.

1.5 Notation

In general, we will use lower-case letters for both scalars and vectors, and capital letters for matrices. The conjugate transpose of a vector x or a matrix A will be denoted x' or A', respectively. For a vector x, its norm is denoted $\|x\|$, and will always refer to the Euclidean norm: $\|x\| = (x'x)^{1/2}$.

All signals will be in discrete time, and we will write $z[k]$ for the value of a signal z at time k. For single-sided signals, the initial sample will be at time $k = 0$. With some abuse of notation, we will use $z[k]$ for both the value of the signal at a particular time k, and the entire sequence $\{z[k], k = 0, 1, \ldots\}$. Signals may be vector or scalar valued. As in MATLAB, $z[a : b]$ denotes $\{z[k]\}_{k=a}^{b}$.

For frequency-domain expressions, z^{-1} will denote the unit delay operator, so a transfer function will be a function of z. The duplicate use of z is to allow the standard notations from Kalman filtering and linear, time-invariant signal processing to co-exist. To write that a signal y is the output of a linear system with input x and transfer function $H(z)$, we will write $y = H(z)x$ or, with even more abuse of notation, $y[k] = H(z)x[k]$.

Chapter 2

Background: Linear Systems, Estimation, Optimization and Quantization

This research combines a number of areas, most importantly state space linear systems theory, LMI optimization, and quantization. The purpose of this chapter is to review some of the elementary facts in these areas. References will be provided throughout for readers interested in greater detail.

Sections 2.1 to 2.3 review elementary concepts in linear state space theory for random processes. Sections 2.4 and 2.5 then cover the equations for the Kalman filter and illustrate the use of the filter for a simple application to denoising. LMI optimization and their application to linear estimation are reviewed in Sections 2.6 and 2.7. Section 2.7 contain the only new results in this chapter: Theorems 2.3 and 2.4. Finally, some basic definitions and properties of scalar quantization are covered in Section 2.8.

2.1 State Space Models for Random Processes

One of the simplest and most widely used models for random processes in signal processing is white noise passed through a linear filter. This work will make considerable use of such models where the linear filter is described in so-called *state space* form. State space representations for systems have been widely used since at least the 1950s and provide compact descriptions that can be analyzed elegantly with linear algebraic methods. Moreover, using standard numerical methods, state space models are particularly amenable to efficient numerical simulation and analysis. State space models are thus ideal for our purpose: designing practical numerical methods for analysis and design that can be applied to a large class of engineering problems.

The current and following two sections will review some elementary results for linear state space systems. All these results are well-known and can be found in any standard text, such as [20, 21, 97]. The presentation here will follow the more modern notation adopted in robust control, such as in [164].

A discrete-time *linear state space system* is simply a set of recursive equations of the form

$$\begin{aligned} x[k+1] &= Ax[k] + Bw[k], \\ z[k] &= Cx[k] + Dw[k], \end{aligned} \tag{2.1}$$

that define a mapping from an input signal $w[k]$ to an output signal $z[k]$. The signal $x[k]$ is an intermediate signal in the mapping and is called the system *state*. All the signals may be vector valued and the terms A, B, C, and D are matrices of the appropriate dimension. With zero initial conditions, the mapping from $w[k]$ to $z[k]$ defines a linear time-invariant (LTI) system whose transfer function is given by,

$$H(z) = C(zI - A)^{-1}B + D.$$

The poles of the transfer function are a subset of the eigenvalues of A. Conversely, any causal, LTI system with a rational transfer function $H(z)$ can be represented in a state space form (2.1), with the dimension of the state vector $x[k]$ equal to the order of the transfer function, and the eigenvalues of A equal to the poles of $H(z)$.

We will often use (2.1) as a model for a random process, where the input $w[k]$ is assumed to be a zero-mean white noise with unit variance, and the output $z[k]$ is the random process to be modeled. Since the mapping from $w[k]$ to $z[k]$ is LTI (with appropriate initial conditions), and the input process $w[k]$ is assumed to be white, the output process $z[k]$ in this model will be wide-sense stationary.

This model is fairly general and can capture the second-order statistics of any stationary random process. To see this, recall that for a stationary random process, the second-order statistics are completely described by the power spectral density. Now, since the input process $w[k]$ in (2.1) is assumed to be white with unit variance, the output process $z[k]$ will have a power spectral density equal to

$$S_{zz}(z) = |H(z)|^2,$$

where $H(z)$ is the transfer function of the mapping from $w[k]$ to $z[k]$ (for the moment, we have assumed that $z[k]$ is a scalar process). By selecting the system matrices (A, B, C, D) corresponding to the appropriate transfer function $H(z)$, we can then produce a random process with an arbitrary rational power spectral density.

For example, if we wish to model a lowpass signal $z[k]$, then we can select the system matrices corresponding to a lowpass filter $H(z)$ with the appropriate bandwidth. Any linear filter design method can be used to select the filter and several techniques are available to represent the filter in state space form. There are also methods for selecting the appropriate filter $H(z)$ and corresponding state space matrices based on empirical data.

Also, since the system (2.1) is equivalent to a linear system with a rational transfer function, the random process model is equivalent to an autoregressive moving average (ARMA) model.

In addition to assuming that the input $w[k]$ is zero-mean and white, it will sometimes be useful to make the additional assumption that the input process is Gaussian. In this case, since the mapping from $w[k]$ to $z[k]$ is linear, the output process $z[k]$ will also be Gaussian. Moreover, since stationary Gaussian random processes are completely described by their second-order statistics, the model (2.1) can completely describe any such Gaussian random process.

Another useful variant of the system in (2.1) is the *linear time-varying* (LTV) system of the form

$$\begin{aligned} x[k+1] &= A[k]x[k] + B[k]w[k], \\ z[k] &= C[k]x[k] + D[k]w[k]. \end{aligned} \tag{2.2}$$

The system (2.2) is identical to (2.1) except that the system matrices, $A[k]$, $B[k]$, etc., are time varying. With zero initial conditions, an LTV system still defines a linear map from $w[k]$ to $z[k]$; however that map is not necessarily time invariant. Consequently, unlike in the LTI case, the mapping cannot be described by a simple transfer function. Also, if the input $w[k]$ is Gaussian, the output $z[k]$ will also be Gaussian, but not necessarily stationary.

2.2 Stability, Stabilizability and Detectability

An important concept in linear systems is stability. In the typical transfer function description of linear systems, stability is usually understood as bounded-input bounded-output, or BIBO, stability. That is, a linear system mapping $w[k]$ to $z[k]$ is considered stable if any bounded input $w[k]$ results in a bounded output $z[k]$. It is well-known that a transfer function is BIBO stable if all its poles are in the open unit circle.

However, for state space models one needs a slightly stronger definition of stability to account for the system's internal state $x[k]$. The typical definition of stability for a state space system is provided by the following simple result:

Proposition 2.1 *For the LTI system (2.1), the following are equivalent:*

(a) The eigenvalues of A are in the open unit circle.

(b) For any initial condition $x[0]$ and bounded input sequence $w[k]$, the resulting state sequence $x[k]$ and output sequence $z[k]$ will be bounded.

(c) For any initial condition $x[0]$,

$$\lim_{k\to\infty} x[k] = 0$$

when $w[k] = 0$ for all k.

Definition 2.1 *An LTI system (2.1) is called* stable *if any of the equivalent conditions of Proposition 2.1 are satisfied.*

From condition (b) of Proposition 2.1, we see that if the state space system is stable in the sense of Definition 2.1, a bounded input $w[k]$ results in a bounded output $z[k]$ *and* bounded state $x[k]$. Since the Definition 2.1 concerns the internal state of the system, this definition of stability is sometimes called *internal stability*, to contrast it against input-output stability. However, in this work, when we use the term "stable" we will always mean stability in the sense of Definition 2.1.

One other point to note is that condition (a) of Proposition 2.1 shows that stability is only a function of the A matrix of the system—the matrices B, C and D have no bearing on the stability. Consequently, we will sometimes drop the reference to the system and say that a matrix A, by itself, is stable if all its eigenvalues are in the open unit circle.

Two other related concepts that we will need are *stabilizability* and *detectability*.

Definition 2.2 *For the state space LTI system (2.1),*

(a) The matrix pair (A, B) is called stabilizable *if there exists an F such that $A - BF$ is stable.*

(b) The matrix pair (A, C) is called detectable *if there exists an L such that $A - LC$ is stable.*

To understand the relevance of stabilizability, suppose that, at each time sample k, $w[k] = -Fx[k]$. Then, the state update in (2.1) would reduce to

$$x[k+1] = (A - BF)x[k].$$

Therefore, if $A - BF$ is stable, the state $x[k] \to 0$. In this sense, the $w[k] = -Fx[k]$ can be understood as a *stabilizing feedback control law*, since it drives the system state to zero from any initial condition. Stabilizability is thus equivalent to the existence of a certain stabilizing feedback.

Detectability has a similar interpretation in the context of state estimation, which we will discuss later.

Both stabilizability and detectability are well-known linear systems concepts, and there are several analytic tests to determine if the conditions are satisfied.

2.3 Lyapunov Equation

Given a time-varying state space model (2.2), it is simple to compute the second-order statistics of the state and output random processes $x[k]$ and $z[k]$. For example, suppose the input process $w[k]$ is zero-mean, white noise with unit variance, and we wish to compute the expected values of the state and output,

$$\hat{x}[k] = \mathbf{E}x[k] \qquad \text{and} \qquad \hat{z}[k] = \mathbf{E}z[k].$$

Let $P[k]$ denote the state covariance matrix

$$P[k] = \mathbf{E}(x[k] - \hat{x}[k])(x[k] - \hat{x}[k])', \tag{2.3}$$

and let $\sigma^2[k]$ denote the mean-squared output,

$$\sigma^2[k] = \mathbf{E}\|z[k] - \hat{z}[k]\|^2. \tag{2.4}$$

Then, it is easily verified that these second-order statistics satisfy the simple recursive relations,

$$\begin{aligned} \hat{x}[k+1] &= A[k]\hat{x}[k], \\ \hat{z}[k] &= C[k]\hat{x}[k], \\ P[k+1] &= A[k]P[k]A[k]' + B[k]B[k]', \\ \sigma^2[k] &= \mathbf{Tr}(C[k]P[k]C[k]' + D[k]D[k]'). \end{aligned} \tag{2.5}$$

For a linear time-invariant system of the form (2.1), one can perform the above recursion with the time-invariant matrices, i.e., $A[k] = A$, $B[k] = B$, etc. As the following proposition indicates, if the system is time invariant and stable, the state and output variance converge.

Proposition 2.2 *Consider the system (2.1) and suppose the input process $w[k]$ is zero-mean, white and has unit variance. Suppose also that A is a stable matrix. Let $\hat{x}[k]$, $\hat{z}[k]$, $P[k]$ and $\sigma^2[k]$ be defined by the recursion (2.5). Then, the covariance matrix $P[k] \to P$, where P is the unique solution to the* Lyapunov equation

$$P = APA' + BB'. \tag{2.6}$$

Also, $\hat{x}[k]$ and $\hat{z}[k] \to 0$ and

$$\lim_{k\to\infty} \sigma^2[k] = \mathbf{Tr}(CPC' + DD'). \tag{2.7}$$

Equation (2.6) is called the *Lyapunov equation* and has been extensively studied. The equation is a linear equation in the matrix P and can be solved efficiently using a variety of numerical methods. In MATLAB, the equation can be solved with the function `dlyap`.

The following proposition gives a useful alternative to Proposition 2.2 for computing the steady-state value for $\sigma^2[k]$.

Proposition 2.3 *Consider the system (2.1). Suppose the input process $w[k]$ is zero-mean, white and has unit variance, and the matrix A is stable. Then, the steady-state mean-squared output is given by*

$$\lim_{k\to\infty} \sigma^2[k] = \mathbf{Tr}(B'QB + D'D),$$

where Q is the unique solution to the Lyapunov equation

$$Q = A'QA + C'C. \tag{2.8}$$

While this alternate form does not provide the state covariance matrix P, we will often use it when we are just interested in the steady-state value for the mean-squared output $\sigma^2[k]$.

As a final note, one important use of the Lyapunov equation is as a test of stability. Specifically, as shown in the following result, under certain conditions, the existence of positive semidefinite solutions to the Lyapunov equation implies stability.

Proposition 2.4 *Consider the system (2.1). Then, A is stable if either of the following conditions is satisfied:*

(a) (A, B) is stabilizable, and there exists a $P \geq 0$ satisfying the Lyapunov equation (2.6).

(b) (A, C) is detectable, and there exists a $Q \geq 0$ satisfying the Lyapunov equation (2.8).

2.4 Kalman State Estimation

2.4.1 Basic Equations

Another basic tool in analyzing stochastic linear systems is the Kalman filter, which provides a simple method for estimating signals described as outputs of linear state space systems. Since we will use the Kalman filter extensively throughout this work, it is worthwhile to briefly review the equations and establish some notation. The Kalman filter originally appeared in [90] and [91], and it is covered in any standard text on linear systems such as [9, 97]. Again, we will omit all derivations.

To describe the Kalman filter, consider the linear state space system of the form

$$\begin{aligned} x[k+1] &= A[k]x[k] + B[k]w[k], \\ z[k] &= C_1[k]x[k] + D_1[k]w[k], \\ y[k] &= C_2[k]x[k] + D_2[k]w[k]. \end{aligned} \tag{2.9}$$

As before, we assume the input $w[k]$ is white noise with zero mean and unit variance. However, unlike the system before, this system has two outputs, $z[k]$ and $y[k]$. The basic Kalman filtering problem is to estimate the first output signal $z[k]$ and state $x[k]$ from the second output $y[k]$. The signal $z[k]$ thus represents an unknown output that we wish to estimate from an observed output $y[k]$.

Define the state and output estimates

$$\hat{x}[k|j] = \mathbf{E}\left(x[k] \mid y[0:j]\right) \qquad \text{and} \qquad \hat{z}[k|j] = \mathbf{E}\left(z[k] \mid y[0:j]\right), \tag{2.10}$$

where $y[0:j]$ represents the subsequence $(y[0], y[1], \ldots, y[j])$. Thus, $\hat{x}[k|j]$ is the MMSE estimate of the state $x[k]$ given the observed output up to sample j. Corresponding to the state estimate $\hat{x}[k|j]$, let $P[k|j]$ denote the estimate error variance

$$P[k|j] = \mathbf{E}\left((x[k] - \hat{x}[k|j])((x[k] - \hat{x}[k|j])' \mid y[0:j]\right). \tag{2.11}$$

For the output estimate, let $\sigma^2[k|j]$ denote the mean-squared output estimation error,

$$\sigma^2[k|j] = \mathbf{E}\|z[k] - \hat{z}[k|j]\|^2.$$

The above MMSE estimates are difficult to compute for an arbitrary distribution on $w[k]$. However, if we make the assumption that $w[k]$ is Gaussian, in addition to being zero mean and white, the MMSE estimates can be described by a simple set of recursive equations. The recursive estimator described by these equations is called the *Kalman filter*.

In the form most useful for our analysis, the Kalman filter can be used to iteratively compute the estimates $\hat{x}[k|k-1]$ and $\hat{z}[k|k-1]$ and state error estimate variance $P[k|k-1]$ with the following recursion:

$$\begin{aligned} e[k] &= y[k] - C_2\hat{x}[k|k-1] \\ \hat{x}[k+1|k] &= A\hat{x}[k|k-1] + L_1[k]e[k] \\ \hat{z}[k|k-1] &= C_1\hat{x}[k|k-1] \\ P[k+1|k] &= AP[k|k-1]A' - E_1[k]G[k]E_1[k]' + BB' \\ E_1[k] &= AP[k|k-1]C_2' + BD_2' \\ G[k] &= (C_2P[k|k-1]C_2' + D_2D_2')^{-1} \\ L_1[k] &= E_1[k]G[k]. \end{aligned} \tag{2.12}$$

To simplify the notation, we have omitted the dependence of the various system matrices on time. For example, we have written A for $A[k]$.

The state estimate update in (2.12) has a simple interpretation. We see that the estimate $\hat{x}[k+1|k]$ is a sum of two terms. It can be verified that the first term, $A\hat{x}[k|k-1]$, is precisely $\hat{x}[k+1|k-1]$, the MMSE estimate of $x[k+1]$ given the observed data up to sample $k-1$. In the second term, the factor $C_2\hat{x}[k|k-1]$ is the MMSE estimate of $y[k]$ given $y[0], \ldots, y[k-1]$, i.e. the *one-step ahead predictor* of $y[k]$. Therefore, the term $y[k] - C_2\hat{x}[k|k-1]$ represents the one-step ahead prediction error. Thus, the state estimate $\hat{x}[k+1|k]$ is the estimate $\hat{x}[k+1|k-1]$ plus a correction term based on the prediction error with the new sample $y[k]$. The multiplicative factor $L_1[k]$ in the correction term is called the *prediction gain matrix.*

The Kalman filter also provides simple formulae for the estimates $\hat{x}[k|k]$ and $\hat{z}[k|k]$:

$$\begin{aligned} \hat{x}[k|k] &= \hat{x}[k|k-1] + P[k|k-1]C_2'G[k]e[k] \\ \hat{z}[k|k] &= C_1\hat{x}[k|k-1] + L_2[k]e[k] \\ L_2[k] &= E_2[k]G[k] \\ E_2[k] &= C_1P[k|k-1]C_2' + D_1D_2'. \end{aligned} \tag{2.13}$$

Finally, the corresponding output mean-squared estimation errors are given by

$$\begin{aligned} \sigma^2[k|k-1] &= \mathbf{Tr}(C_1P[k|k-1]C_1' + D_1D_1') \\ \sigma^2[k|k] &= \mathbf{Tr}(C_1P[k|k-1]C_1' + D_1D_1' - E_2[k]G[k]E_2[k]). \end{aligned} \tag{2.14}$$

An important property of the Kalman filter is that the MMSE estimates $\hat{x}[k|k-1]$ and $\hat{z}[k|k-1]$ are linear functions of the observed data $y[k]$. This linearity of the optimal estimator is a result of the assumption that $w[k]$ is Gaussian. If we relax the assumption that $w[k]$ is Gaussian, but still assume it is zero mean and white with unit variance, then it turns out that the Kalman filter is the optimal *linear* estimator.

2.4.2 Systems with Known Inputs

In the filtering problem above, the only input was the unknown noise $w[k]$. However, in many problems there may also be known components in the input and output of the system. Fortunately, such known inputs are relatively easy to handle by "subtracting out" their effect.

To describe the Kalman filtering problem with known inputs, consider the system

$$\begin{aligned} x[k+1] &= A[k]x[k] + B[k]w[k] + u_0[k] \\ z[k] &= C_1[k]x[k] + D_1[k]w[k] + u_1[k] \\ y[k] &= C_2[k]x[k] + D_2[k]w[k] + u_2[k], \end{aligned} \tag{2.15}$$

which is identical to the LTV system in (2.9) with the addition of the signals $u_i[k]$, $i = 0, 1, 2$. The signals $u_i[k]$ represent signals independent of the noise and known to the estimator. We wish to find estimates of the state $x[k]$ and desired output $z[k]$ from the measured output $y[k]$ and the known signals $\{u_i[k]\}_{i=0}^2$. We denote these estimates by

$$\hat{x}[k|j] = \mathbf{E}\left(x[k] \mid y[0:j], u[0:j]\right) \qquad \text{and} \qquad \hat{z}[k|j] = \mathbf{E}\left(z[k] \mid y[0:j], u[0:j]\right),$$

where $u[k] = (u_0[k], u_1[k], u_2[k])$ is a vector of known input signals. The state and output estimates are identical to before, except that they depend also on the known inputs. Also, as before, we can define the state error variance, $P[k|j]$, and mean-squared output variance, $\sigma^2[k|j]$. We will assume that, for every k, $w[k]$ is independent of all $u[j]$ for $j \le k$.

Under these assumptions, the estimator equations are identical to (2.12) and (2.13) given earlier, except for the state and output estimate updates:

$$\begin{aligned} e[k] &= y[k] - C_2\hat{x}[k|k-1] - u_1[k] \\ \hat{x}[k+1|k] &= A\hat{x}[k|k-1] + L_1[k]e[k] + u_0[k] \\ \hat{z}[k|k] &= C_1\hat{x}[k|k-1] + L_2[k]e[k] + u_2[k] \end{aligned} \tag{2.16}$$

We see here that the known inputs are handled by "subtracting out" the signal $u_2[k]$ from $y[k]$, and then adding the input and output signals $u_0[k]$ and $u_1[k]$ into the updates for $\hat{x}[k|k]$ and $\hat{z}[k|k]$. There is no change in the updates for the error variance $P[k|k-1]$, the mean-squared output $\sigma^2[k|k]$, or the gain matrices $L_1[k]$ and $L_2[k]$. In particular, the addition of known signals does not affect the expected estimation error.

2.4.3 Algebraic Riccati Equation and the Steady-State Filter

The Kalman filter updates in (2.12) and (2.13), in general, produce a time-varying sequence of error covariance matrices $P[k|k-1]$ and gain matrices $L_1[k]$ and $L_2[k]$. However, if the system is time-invariant, then under mild assumptions the filter parameters eventually converge as stated in the following theorem.

Theorem 2.1 *Consider the Kalman filter recursion in (2.12) and (2.13) applied to a time-invariant system. Suppose that* (A, C_2) *is detectable and* $D_2 D_2' > 0$. *Then the error covariance matrix* $P[k|k-1] \rightarrow P$ *where* P *is the unique positive semidefinite solution to the* algebraic Riccati equation*:*

$$P = APA' + BB' - E_1 G E_1', \qquad \text{where} \qquad \begin{aligned} E_1 &= APC_2' + BD_2', \\ G &= (C_2 P C_2' + D_2 D_2')^{-1}. \end{aligned} \tag{2.17}$$

Also, the Kalman gain matrices $L_1[k] \rightarrow L_1 = E_1 G$ *and* $L_2[k] \rightarrow L_2 = E_2 G$, *where* $E_2 = C_1 P C_2' + D_1 D_2'$. *Finally, the asymptotic mean-squared errors are given by*

$$\begin{aligned} \lim_{k \to \infty} \sigma^2[k|k-1] &= \mathbf{Tr}\left(C_1 P C_1' + D_1 D_1' \right), \\ \lim_{k \to \infty} \sigma^2[k|k] &= \mathbf{Tr}\left(C_1 P C_1' + D_1 D_1' - E_2 G E_2 \right). \end{aligned}$$

The properties of the algebraic Riccati equation (2.17) were originally studied in the fundamental paper of Wonham [160]. Since then, the equation has been extensively studied, and it can be efficiently solved by simple numerical algorithms. In MATLAB, the equation can be solved with the function `dare`.

If we substitute the limiting gains L_1 and L_2 from Theorem 2.1 into the Kalman filter equations (2.12) and (2.13), we obtain the so-called *steady-state Kalman filter*:

$$\begin{aligned} \hat{x}[k+1|k] &= A\hat{x}[k|k-1] + L_1(y[k] - C_2 \hat{x}[k|k-1]) \\ \hat{z}[k|k-1] &= C_1 \hat{x}[k|k-1] \\ \hat{z}[k|k] &= C_1 \hat{x}[k|k-1] + L_2(y[k] - C_2 \hat{x}[k|k-1]). \end{aligned} \tag{2.18}$$

Computationally, the steady-state filter is significantly simpler than the time-varying filter in (2.12). For the steady-state filter, one must initially solve the algebraic Riccati equation to compute the error covariance matrix P and the corresponding gain matrices L_1 and L_2. But, then at runtime, the covariance matrix updates for $P[k|k-1]$ in (2.12) do not need to be computed, greatly reducing the per-sample computation. Indeed, with $L_1[k]$ and $L_2[k]$ set to constants, the steady-state Kalman filter (2.18) is itself a linear time-invariant filter.

As the following theorem states, for time-invariant systems, the steady-state Kalman filter is *asymptotically optimal*. That is, it achieves the same mean-squared error as the optimal time-varying Kalman filter. The only possible disadvantage is in the initial performance, as dictated by the initial conditions.

Theorem 2.2 *Consider the steady-state Kalman filter (2.18) used with the time-invariant system (2.9). Assume that* (A, C_2) *is detectable,* $D_2 D_2' > 0$*, and* L*,* G *and* E *are computed as in Theorem 2.1. Then, the asymptotic mean-squared error of the steady-state filter is identical to that of the time-varying filter. That is,*

$$\lim_{k\to\infty} \|z[k] - z[k|k-1]\|^2 = \mathbf{Tr}\left(C_1 P C_1' + D_1 D_1'\right)$$

and

$$\lim_{k\to\infty} \|z[k] - z[k|k]\|^2 = \mathbf{Tr}\left(C_1 P C_1' + D_1 D_1' - E_2 G E_2\right).$$

2.4.4 One-Step Ahead Prediction

One important use of the Kalman filter is for *one-step ahead prediction*, where we are to estimate samples of a random process $z[k]$ from past samples $z[j]$ up to time $j = k-1$. We will see, in Chapter 5, that the one-step ahead prediction problem plays an important role in the design of encoders for predictive quantization.

To pose the one-step ahead prediction problem as a Kalman filtering problem, we first suppose that $z[k]$ can be written as the output of a state space system, (2.1), where $w[k]$ is zero-mean, unit-variance white noise. We next define $y[k] = z[k]$ to obtain the state space system

$$\begin{aligned} x[k+1] &= Ax[k] + Bw[k] \\ z[k] &= Cx[k] + Dw[k] \\ y[k] &= Cx[k] + Dw[k], \end{aligned} \tag{2.19}$$

which is identical to the system (2.9) in the Kalman filter problem with

$$C_1 = C_2 = C \qquad \text{and} \qquad D_1 = D_2 = D.$$

Now, define the one-step ahead predictors of the desired signal $z[k]$ and the state $x[k]$ as

$$\begin{aligned} \hat{z}[k|k-1] &= \mathbf{E}\left(z[k] \mid z[k-1], z[k-2], \ldots\right) \\ \hat{x}[k|k-1] &= \mathbf{E}\left(x[k] \mid z[k-1], z[k-2], \ldots\right), \end{aligned}$$

and let $P[k|k-1]$ and $\sigma^2[k]$ be corresponding quantities:

$$P[k|k-1] = \mathbf{E}\left((x[k] - \hat{x}[k|k-1])(x[k] - \hat{x}[k|k-1])'\right)$$

and

$$\sigma^2[k] = \mathbf{E}\|z[k] - \hat{z}[k|k-1]\|^2.$$

Since $y[k] = z[k]$, these estimates are precisely the outputs of a Kalman filter. For the prediction problem, where $C_1 = C_2 = C$ and $D_1 = D_2 = D$, the asymptotic Kalman filter (2.18) reduces to

$$\begin{aligned} \hat{x}[k+1|k] &= A\hat{x}[k|k-1] + L(z[k] - C\hat{x}[k|k-1]) \\ \hat{z}[k|k-1] &= C\hat{x}[k|k-1], \end{aligned} \tag{2.20}$$

where L is the Kalman filter gain given by $L = E_1 G$, and E_1 and G are given by the solution to the algebraic Riccati equation,

$$P = APA' + BB' - E_1 G E_1', \qquad \text{where} \qquad \begin{aligned} E_1 &= APC' + BD' \\ G &= (CPC' + DD')^{-1}. \end{aligned} \tag{2.21}$$

Moreover, the asymptotic prediction errors are given by

$$\lim_{k\to\infty} \sigma^2[k|k-1] = \mathbf{Tr}\left(CPC' + DD'\right)$$

and

$$\lim_{k\to\infty} P[k|k-1] = P.$$

We thus see that if the random process $z[k]$ can be described as the output of a linear state space system, its one-step ahead predictor is given by a simple Kalman filter. Moreover, the solution P to the algebraic Riccati equation (2.21) provides the asymptotic one-step ahead state prediction error variance.

2.4.5 Noncausal Estimation

Up to now we have considered only causal estimation, where the estimates of $z[k]$ and $x[k]$ depend only on data $y[j]$ up to the time $j = k$. However, the Kalman filter can also be used for non-causal estimation. In this subsection, we review a standard non-causal estimation problem known as *fixed-interval smoothing*. In this case, we are given a finite sequence of data $y[0], \ldots, y[T]$, and we are interested in estimating each state sample $x[k]$ and output sample $z[k]$ from the entire observed sequence $y[0], \ldots, y[T]$. In the notation above, we want to compute $\hat{x}[k|T]$ and $\hat{z}[k|T]$ and the corresponding errors.

The solution to the fixed-interval smoothing problem has a well-known two-pass solution that was originally derived by Mayne [111]. In the first pass, the standard causal Kalman filter is used to obtain the causal estimates $\hat{x}[k|k]$ and $\hat{z}[k|k]$ as before.

In the second pass, a Kalman filter is run in *reverse time* to improve the estimates:

$$\begin{aligned}
v[k-1] &= A_L[k]'v[k] + C_2[k]'G[k]e[k] \\
\hat{x}[k|T] &= \hat{x}[k|k] + P[k|k-1]A_L[k]'v[k] \\
\hat{z}[k|T] &= \hat{z}[k|k] + P_1[k]'v[k] \\
Q[k-1] &= A_L[k]'Q[k]A_L[k] + C_2[k]'G[k]C_2[k] \\
P_1[k] &= AP[k|k-1]C_1' + BD_1' - E_1[k]G[k]E_2[k]' \\
P[k|T] &= P[k|k] - P[k|k-1]A_L[k]'Q[k]A_L[k]P[k|k-1] \\
\sigma^2[k|T] &= \sigma^2[k|k] - \mathbf{Tr}(P_1[k]'Q[k]P_1[k]) \\
A_L[k] &= A[k] - L_1[k]C_2[k]
\end{aligned} \tag{2.22}$$

with the initial conditions $v[T] = 0$ and $Q[T] = 0$. The filter (2.22) is initialized at $k = T$ and then stepped in reverse until the final sample $k = 0$. Similar to Section 2.4.2, known inputs can also be considered.

Similar to Section 2.4.3, we can also consider the asymptotic performance. For non-causal estimation, the asymptotic performance would be measured at a point k both far from the initial sample at time 0 and the terminal point T. Mathematically, this limit could be written as

$$\begin{aligned}
P_\infty &= \lim_{k\to\infty} \lim_{T\to\infty} P[k|T] \\
\sigma^2_\infty &= \lim_{k\to\infty} \lim_{T\to\infty} \sigma^2[k|T].
\end{aligned}$$

Using the reverse update equations (2.22), it can be shown these limits are given by

$$\begin{aligned}
P_\infty &= P - E_1GE_1' - PA_L'QA_LP \\
\sigma^2_\infty &= \mathbf{Tr}\left(C_1PC_1' + D_1D_1' - E_2GE_2 - P_1'QP_1\right),
\end{aligned}$$

where P, E_1, E_2 and G are given in Section 2.4.3, $A_L = A - L_1C_2$, $Q \geq 0$ is the solution to the Lyapunov equation

$$Q = A_L'QA_L + C_2'GC_2, \tag{2.23}$$

and $P_1 = APC_1 + BD_1' - E_1GE_2'$. Thus, the asymptotic smoothing performance is described by the solution P to the algebraic Riccati equation (2.17) along with the Lyapunov equation (2.23).

2.5 Denoising with Kalman Filtering

A common application of Kalman filtering is optimal filter design for signal denoising. Suppose we wish to estimate a signal $z[k]$ from a noisy version $y[k]$ given by

$$y[k] = z[k] + d[k], \tag{2.24}$$

where $d[k]$ is additive noise. The estimation of $z[k]$ from $y[k]$ is often called *denoising*, since the problem involves removing the noise $d[k]$ from the observed signal $y[k]$.

To pose the denoising problem as a Kalman filtering problem, we assume that $z[k]$ is a Gaussian random process that can be described as the output of state space system

$$\begin{aligned} x[k+1] &= A_z x[k] + B_z w_z[k] \\ z[k] &= C_z x[k] + D_z w_z[k], \end{aligned} \tag{2.25}$$

where $w_z[k]$ is zero-mean, Gaussian white noise with unit variance. As before, the system matrices (A_z, B_z, C_z, D_z) should be chosen to match the statistics of $z[k]$. For example, if $z[k]$ is lowpass, then the matrices can correspond to a lowpass filter with the appropriate signal bandwidth.

For the noise $d[k]$, we will assume it can be written as

$$d[k] = \sigma_d^2 w_d[k], \tag{2.26}$$

where $w_d[k]$ is also zero-mean, Gaussian, unit-variance white noise. The factor σ_d^2 is a scaling term to account for the noise variance.

Now, combining (2.24), (2.25) and (2.26), we can write

$$\begin{aligned} x[k+1] &= Ax[k] + Bw[k] \\ z[k] &= C_1 x[k] + D_1 w[k] \\ y[k] &= C_2 x[k] + D_2 w[k], \end{aligned} \tag{2.27}$$

where $w[k] = [w_z[k]' \;\; w_d[k]']'$ is a noise vector containing both the source noise $w_z[k]$ and additive noise $w_d[k]$, and the system matrices in (2.27) are given by

$$\begin{aligned} A &= A_z, & B &= [B_z \;\; 0], \\ C_1 &= C_z, & D_1 &= [D_z \;\; 0], \\ C_2 &= C_z, & D_2 &= [D_z \;\; \sigma_d^2]. \end{aligned}$$

With these definitions, the system (2.27) is in the form of (2.9), and we can apply the Kalman filter to find the optimal MMSE estimate of $z[k]$ given $y[k]$. The Kalman filter can be used to find the strictly causal, causal or non-causal estimate.

2.6 Linear Matrix Inequalities (LMIs)

Linear Matrix Inequality (LMI) optimization is a type of convex optimization that became popular in the 1990s in the analysis of robust control systems. The LMI optimization problem can be seen as a matrix-valued form of the standard linear programming problem, and it can be solved efficiently with similar methods. While LMIs were initially used in robust control, they have since proven to be useful in linear systems problems in numerous areas. We will make extensive use of LMI optimization in the analysis of estimation problems considered in this work. Here, we will present some simple definitions for LMI optimization. The interested reader is referred to the book [16] for a comprehensive treatment of the subject.

The basic LMI optimization problem can be described as

$$\min_x f(x)$$

subject to

$$g_i(x) \geq 0, \quad i = 1, \ldots, M,$$

where the decision variable x is a vector, and the objective function $f(x)$ and constraint functions $g_i(x)$ are affine functions of x. The problem is thus identical to the standard linear programming problem, except that in the LMI problem, the constraint functions $g_i(x)$ are permitted to be *matrix* valued. That is, each constraint g_i is a mapping from the vector x to the real-valued matrices $\mathbb{R}^{n \times n}$ for some dimension n. The dimension of the output matrix must be constant and the coefficients of the matrix must be affine functions of the input x. By the constraint, $g_i(x) \geq 0$, we mean that the matrix $g_i(x)$ must be symmetric and positive semidefinite. The objective function $f(x)$ is scalar valued.

In the special case where the constraints $g_i(x)$ are scalar valued, the LMI problem reduces to the standard linear programming problem. However, by allowing matrix-valued constraints, the LMI problem is considerably more general. As we will see, the problem encompasses quadratic programming problems and several quadratic matrix type problems that arise in estimation and control.

Since the set of positive semidefinite matrices is convex, and the objective and constraint functions in the LMI problem are affine, the LMI optimization is itself convex. Consequently, the optimization problem can, in general, be solved for a global minimum relatively easily. In fact, there are currently several commercially-available LMI solvers that can efficiently solve LMI optimization problem with hundreds of variables and constraints. Most solvers are based on a modified of form of the ellipsoidal algorithm used in Karmarkar's method for linear programming. In this work, we will use the MATLAB package `lmitool`, which provides a convenient and simple interface.

A useful identity that arises often in LMI analysis is called the *Schur complement* and described in the following lemma.

Lemma 2.1 *Consider a matrix P of the form*

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22}, \end{bmatrix}$$

where the terms P_{ij} are submatrices. Suppose $P = P'$ and $P_{22} > 0$. Then $P \geq 0$ if and only if

$$P_{11} - P_{12}P_{22}^{-1}P_{21} \geq 0. \tag{2.28}$$

As a typical example of how the Schur complement is used, consider the well-known quadratic programming problem, of minimizing the quadratic function

$$f_0(x) = x'Qx + c'x + d, \tag{2.29}$$

subject to the linear constraints

$$Ax \leq b,$$

where A and Q are matrices, b and c are row vectors, d is a scalar, and $Q > 0$. Due to the quadratic term, $x'Qx$, the objective function cannot directly be used in an LMI. However, observe that for any γ, $f_0(x) \leq \gamma$ if and only if

$$x'Qx + c'x + d \leq \gamma.$$

By the Schur complement, this condition is in turn equivalent to the LMI constraint,

$$g(x, \gamma) = \begin{bmatrix} \gamma - c'x - d & x' \\ x & Q^{-1} \end{bmatrix} \geq 0.$$

Therefore, the quadratic minimization (2.29) is equivalent to

$$\min_{x,\gamma} \gamma$$

subject to

$$\begin{aligned} g(x,\gamma) &\geq 0 \\ Ax &\leq b \end{aligned}$$

Now, all the submatrices in $g(x,\gamma)$ are jointly affine in x and γ. Therefore, the quadratic programming problem has been translated to a linear objective with an additional affine matrix constraint, and can thus be solved as an LMI. Of course, it is more efficient to solve quadratic programming problems with a specialized quadratic program solver rather as a more general LMI problem. But, the example illustrates how LMIs typically arise.

2.7 LMI Approach to Linear Estimation

LMIs are particularly useful in analyzing the random linear systems introduced in Section 2.1. Both the Lyapunov analysis in Section 2.3 and the Kalman estimation problem in Section 2.4 can be formulated with LMIs. The following subsection presents an LMI formulation of the Lyapunov analysis. The Kalman estimation will be considered in Sections 2.7.2 and 2.7.3.

2.7.1 Lyapunov Analysis

Consider again the linear system in (2.1), and suppose the input $w[k]$ is a zero-mean, white noise process. Suppose we wish to compute the asymptotic state covariance and mean-squared output. Propositions 2.2 and 2.3 show that, if the system is stable, the asymptotic values can be computed by the Lyapunov equation. The following proposition shows that the quantities can also be estimated by a related LMI.

Proposition 2.5 *Consider the system (2.1) and suppose the input process $w[k]$ is zero-mean, white and has unit variance. Let $P[k]$ and $\sigma^2[k]$ be the state covariance and mean-squared output defined in (2.3) and (2.4).*

(a) Suppose that (A,B) is stabilizable and $P \geq 0$ satisfies the Lyapunov LMI

$$P \geq APA' + BB'. \tag{2.30}$$

Then, the matrix A is stable and the asymptotic values for the covariance matrix and mean-squared output are bounded by

$$\lim_{k\to\infty} P[k] \leq P$$

and

$$\lim_{k\to\infty} \sigma^2[k] \leq \mathbf{Tr}(CPC' + DD').$$

(b) *Suppose that (A, C) is detectable and $Q \geq 0$ satisfies the Lyapunov LMI*

$$Q \geq A'QA + C'C. \tag{2.31}$$

Then, the matrix A is stable and the asymptotic mean-squared output is bounded by

$$\lim_{k\to\infty} \sigma^2[k] \leq \mathbf{Tr}(B'QB + D'D).$$

Proposition 2.5(a) and (b) can be seen as LMI versions of Propositions 2.2 and 2.3. Specifically, the Lyapunov equation in (2.6) and the expression (2.7) for the mean-squared output are replaced by inequalities in Proposition 2.5(a). Proposition 2.5(b) and Proposition 2.3 are similarly related.

Now, combining Proposition 2.2 and Proposition 2.5(a), we see that the asymptotic mean-squared output variance is given by the minimization

$$\lim_{k\to\infty} \sigma^2[k] = \min_P \mathbf{Tr}(CPC' + DD'), \tag{2.32}$$

subject to the constraints

$$\begin{aligned} P &\geq APA' + BB', \\ P &\geq 0. \end{aligned} \tag{2.33}$$

The objective function (2.32) and constraint functions (2.33) in this minimization are affine functions of the variable P. Also, the objective function is scalar-valued and the constraints are matrix-valued. The minimization is thus an LMI in the variable P that can be used to compute the asymptotic mean-squared output.

However, while this LMI minimization is relatively easy to solve, it is still computationally simpler to directly compute the solution to the Lyapunov equation in Proposition 2.2. For our purposes, the LMI formulation will be useful as an analytical tool that can be extended to more complicated problems, such as jump linear systems that we consider in the next chapter.

2.7.2 Strictly Causal Kalman Filtering

The steady-state state estimation problem can also formulated as an LMI. Consider the linear system (2.9) and an estimator of the form

$$\begin{aligned} \hat{x}[k+1] &= A\hat{x}[k] + L(y[k] - C_2\hat{x}[k]) \\ \hat{z}[k] &= C_1\hat{x}[k] \end{aligned} \tag{2.34}$$

for some gain matrix L. Observe that this estimator for $z[k]$ is *strictly causal*. That is, $\hat{z}[k]$ depends only on samples $y[j]$ up to time $j = k-1$. We will consider the causal estimator in the next subsection.

If we take $L = L_1$, where L_1 is the Kalman gain matrix in Theorem 2.1, the estimator (2.34) reduces to the Kalman filter and achieves the minimum asymptotic mean-squared estimation error. In the LMI formulation, we treat L as a design parameter and optimize it to find the minimum error. That is, we attempt to minimize the *asymptotic mean-squared error*, defined as

$$\sigma^2(L) = \lim_{k\to\infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2, \tag{2.35}$$

where the dependence on L is through the estimator $\hat{z}[k]$.

Before stating the main result, we will need to impose one technical condition. We will say that a gain matrix L is *stabilizing* if the matrix $A - LC_2$ is stable. From Definition 2.2, the existence of such a matrix L is precisely the definition of (A, C_2) being detectable. The relevance of this detectability to the estimation is given by the following simple proposition.

Proposition 2.6 *Consider the LTI system (2.9) and estimator (2.34). Then, a matrix L is stabilizing if and only if for any initial condition $(x[0], \hat{x}[0])$,*

$$\lim_{k\to\infty} x[k] - \hat{x}[k] = 0$$

when $w[k] = 0$ for all k.

Proof: When $w[k] = 0$, we can subtract the state update equations in (2.9) and (2.34) to obtain,

$$x[k+1] - \hat{x}[k+1] = (A - LC_2)(x[k] - \hat{x}[k]).$$

Therefore, $x[k] - \hat{x}[k] \to 0$ if and only if $A - LC_2$ is stable.

Proposition 2.6 shows that a gain matrix L being stabilizing is equivalent to the state estimation error decaying to zero whenever there is no noise. This is a reasonable condition to impose on any estimator, and therefore we will thus focus only on stabilizing gain matrices.

With these preliminary definitions, we can state the main result.

Theorem 2.3 *Consider the linear system (2.9), where the input $w[k]$ is white noise with unit variance. For a given gain matrix L, let $\hat{z}[k]$ be the estimator defined in (2.34) and let $\sigma^2(L)$ be the asymptotic mean-squared error in (2.35).*

(a) Suppose C_1 is injective and there exists a matrix $W \geq 0$, partitioned as

$$W = \begin{bmatrix} W_1 & W_2 \\ W_2' & W_3 \end{bmatrix},$$

that satisfies

$$W_1 \geq [A' \;\; C_2']W[A' \;\; C_2']' + C_1'C_1. \tag{2.36}$$

Then, $W_1 > 0$. Also, if we define

$$L = -W_1^{-1}W_2, \tag{2.37}$$

the matrix L is stabilizing and the asymptotic mean-squared error is bounded by

$$\sigma^2(L) \leq \mathbf{Tr}\left([B' \;\; D_2']W[B' \;\; D_2']') + D_1'D_1\right). \tag{2.38}$$

(b) Conversely, for any stabilizing matrix L, there exists a $W \geq 0$ satisfying (2.36) with

$$\mathbf{Tr}\left([B' \;\; D_2']W[B' \;\; D_2']') + D_1'D_1\right) \leq \sigma^2(L). \tag{2.39}$$

Before proving the theorem, it is worthwhile to briefly discuss the assumption in part (a) that C_1 is injective, since this sort of assumption will be common. At first, it may appear that the assumption is somewhat strong. For example, it requires that $z[k]$ has a dimension greater than or equal to $x[k]$. While the assumption may not hold for some systems, any system can be "perturbed" a small amount to satisfy the assumption. Specifically, suppose that C_1 is not injective. Then, we can consider a new system where C_1 and D_1 are replaced by $C_1^{new} = [C_1' \;\; \epsilon I]'$ and $D_1^{\rm new} = [D_1' \;\; 0]'$ respectively, for some small $\epsilon > 0$. Then, $C_1^{\rm new}$ is injective and the assumption is satisfied.

Proof of Theorem 2.3: For any estimator of the form (2.34), define the error signals

$$e_x[k] = x[k] - \hat{x}[k] \qquad \text{and} \qquad e_z[k] = z[k] - \hat{z}[k].$$

Then, subtracting (2.9) and (2.34), we obtain,

$$\begin{aligned} e_x[k+1] &= (A - LC_2)e_x[k] + (B - LD_2)w[k] \\ e_z[k] &= C_1 e_x[k] + D_1 w[k]. \end{aligned}$$

In this representation, we see that the error signal $e_z[k]$ is expressed as the output of a new linear system with $w[k]$ as the input. The mapping from $w[k]$ to $e_z[k]$ is often called the *closed-loop system*. Since $e_x[k] = x[k] - \hat{x}[k]$, we see that the matrix L is stabilizing if and only if the closed-loop matrix $A - LC_2$ is stable. Also, by the definition of $e_z[k]$,

$$\sigma^2(L) = \lim_{k\to\infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2 = \lim_{k\to\infty} \mathbf{E}\|e_z[k]\|^2.$$

Now, to prove part (a), suppose there exists a matrix $W \geq 0$ satisfying (2.36). First observe that since C_1 is injective, $C_1'C_1 > 0$. Also, since $W \geq 0$, it follows from (2.36) that $W_1 > 0$. Therefore, the inverse W_1^{-1} exists, and we can define L as in (2.37). Now, if we let $Q = W_1$ and define L as in (2.37), we see that

$$\begin{aligned} &(A - LC_2)'Q(A - LC_2) + C_1'C_1 \\ &= \; A'QA - C_2'L'QA - A'QLC_2 + C_2'L'QLC_2 + C_1'C_1 \\ &= \; A'W_1A + C_2'W_2'A + A'W_2C_2 + C_2'W_2'W_1^{-1}W_2C_2 + C_1'C_1 \\ &\leq \; A'W_1A + C_2'W_2'A + A'W_2C_2 + C_2'W_3C_2 + C_1'C_1 \\ &= \; [A' \;\; C_2']W[A' \;\; C_2']' + C_1'C_1 \\ &\leq \; W_1 = Q. \end{aligned}$$

where we have used the Schur complement relation that

$$W_3 \geq W_2'W_1^{-1}W_2.$$

This identity shows that Q satisfies the Lyapunov LMI for the closed-loop system. Also, since C_1 is injective, $(A - LC_2, C_1)$ is detectable. Therefore, by Proposition 2.5(b), the closed-loop matrix $A - LC_2$ must be stable and consequently L is stabilizing. Moreover, the asymptotic mean-squared error is bounded by

$$\begin{aligned}
\sigma^2(L) &\leq \mathbf{Tr}\left((B - LD_2)'Q(B - LD_2) + D_1'D_1\right) \\
&= \mathbf{Tr}\left((B'QB - D_2L'QB - B'QLD_2 + D_2'L'QLD_2 + D_1'D_1\right) \\
&= \mathbf{Tr}\left((B'W_1B + D_2W_2B + B'W_2D_2 + D_2'W_2W_1^{-1}W_2D_2 + D_1'D_1\right) \\
&\leq \mathbf{Tr}\left([B'W_1B + D_2W_2B + B'W_2D_2 + D_2'W_3D_2 + D_1'D_1\right) \\
&= \mathbf{Tr}\left([B' \;\; D_2']W[B' \;\; D_2']' + D_1'D_1\right),
\end{aligned}$$

which proves part (a).

Conversely, consider any stabilizing matrix L. Since L is stabilizing, the closed-loop matrix $A - LC_2$ must be stable. Consequently, by Proposition 2.3, there exists a $Q \geq 0$ satisfying the Lyapunov equation for the closed-loop system,

$$Q = (A - LC_2)'Q(A - LC_2) + C_1'C_1, \tag{2.40}$$

with

$$\sigma^2(L) = \mathbf{Tr}\left((B - LD_2)'Q(B - LD_2) + D_1'D_1\right). \tag{2.41}$$

If we define

$$W_1 = Q, \quad W_2 = -QL, \quad \text{and} \quad W_3 = L'QL, \tag{2.42}$$

we see that,

$$W = \begin{bmatrix} Q & -QL \\ -L'Q & L'QL \end{bmatrix} = [I \;\; -L]'Q[I \;\; -L] \geq 0,$$

so $W \geq 0$. Also, using (2.42) along with (2.40), it can be verified that W satisfies the LMI (2.36). Combining (2.42) and (2.41) shows that W satisfies the bound in (2.39). This proves part (b).

2.7.3 Causal Kalman Filtering

The estimator (2.34) in the previous subsection was strictly causal. We can also use LMIs to find the optimal causal estimator. To this end, consider again the linear system (2.9) and an estimator of the form

$$\begin{aligned}
\hat{x}[k+1] &= A\hat{x}[k] + L_1(y[k] - C_2\hat{x}[k]) \\
\hat{z}[k] &= C_1\hat{x}[k] + L_2(y[k] - C_2\hat{x}[k])
\end{aligned} \tag{2.43}$$

for some matrices L_1 and L_2. This estimator is causal since the estimate $\hat{z}[k]$ for $z[k]$ depends on samples $y[j]$ up to time $j = k$.

Again, the optimal steady-state gain matrices L_1 and L_2 are given by Theorem 2.1. However, for the LMI formulation, we treat L_1 and L_2 as design parameters that are optimized to minimize the asymptotic error, defined as

$$\sigma^2(L_1, L_2) = \lim_{k\to\infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2, \tag{2.44}$$

where the dependence of L_1 and L_2 is through the estimate $\hat{z}[k]$. As in the strictly causal estimator case, the optimal L_1 and L_2 can be found by an LMI.

Theorem 2.4 *Consider the linear system (2.9), where the input $w[k]$ is white noise with unit variance. Given matrices L_1 and L_2, let $\hat{z}[k]$ be the estimator defined in (2.43) and let $\sigma^2(L_1, L_2)$ be the asymptotic mean-squared error in (2.44).*

(a) Suppose that $[C_1' \;\; C_2']'$ is injective and there exist matrices $W \geq 0$ and $V \geq 0$, of the form

$$W = \begin{bmatrix} W_1 & W_2 \\ W_2' & W_3 \end{bmatrix}, \quad V = \begin{bmatrix} I & V_2 \\ V_2' & V_3 \end{bmatrix}, \tag{2.45}$$

that satisfy

$$W_1 \geq [A' \;\; C_2']W[A' \;\; C_2']' + [C_1' \;\; C_2']V[C_1' \;\; C_2']'. \tag{2.46}$$

Then $W_1 > 0$. Also, if we define

$$L_1 = -W_1^{-1}W_2 \qquad \text{and} \qquad L_2 = -V_2, \tag{2.47}$$

the matrix L_1 is stabilizing and the asymptotic mean-squared error is bounded by

$$\sigma^2(L_1, L_2) \leq \mathbf{Tr}\left([B' \;\; D_2']W[B' \;\; D_2']') + [D_1' \;\; D_2']V[D_1' \;\; D_2']'\right). \tag{2.48}$$

(b) Conversely, for any stabilizing matrices L_1 and L_2, there exist matrices $W \geq 0$ and $V \geq 0$ of the form (2.45) satisfying (2.46) with

$$\mathbf{Tr}\left([B' \;\; D_2']W[B' \;\; D_2']') + [D_1' \;\; D_2']V[D_1' \;\; D_2']'\right) \leq \sigma^2(L_1, L_2). \tag{2.49}$$

Proof: The proof is almost identical to that of Theorem 2.3, so we will just outline the proof in this case. Defining the signals $e_x[k]$ and $e_z[k]$ as in the proof of Theorem 2.3, we obtain the closed-loop state space system

$$\begin{aligned} e_x[k+1] &= (A - L_1 C_2) e_x[k] + (B - L_1 D_2) w[k] \\ e_z[k] &= (C_1 - L_2 C_2) e_x[k] + (D_1 - L_2 D_2) w[k]. \end{aligned}$$

Also, the asymptotic mean-squared error is

$$\sigma^2(L_1, L_2) = \lim_{k\to\infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2 = \lim_{k\to\infty} \mathbf{E}\|e_z[k]\|^2.$$

Now, suppose there exists a matrix $W \geq 0$ and $V \geq 0$ as in part (a) of the Theorem. We will first prove that $W_1 > 0$. Observe that

$$\begin{aligned} &[C_1' \;\; C_2']' V [C_1' \;\; C_2']' \\ &= C_1' C_1 + C_1' V_2 C_2 + C_2' V_2' C_1 + C_2' V_3 C_2 \\ &= (C_1 + V_2 C_2)'(C_1 + V_2 C_2) + C_2'(V_3 - V_2' V_2) C_2 \\ &\geq (C_1 + V_2 C_2)'(C_1 + V_2 C_2), \end{aligned}$$

where in the last step we have use the Schur complement on the matrix $V \geq 0$ to show that $V_3 \geq V_2' V_2$. Also, note that

$$C_1 + V_2 C_2 = [I \;\; V_2][C_1' \;\; C_2']'.$$

Now, by the assumption of the theorem, $[C_1' \;\; C_2']'$ is injective, as is the matrix $[I \;\; V_2]$. Since the product of injective matrices is injective, $C_1 + V_2 C_2$ is injective, and therefore, $(C_1 + V_2 C_2)'(C_1 + V_2 C_2) > 0$. It follows from (2.50) that

$$[C_1' \;\; C_2']' V [C_1' \;\; C_2'] > 0.$$

Combining this with the fact that $W \geq 0$, (2.46) shows that $W_1 > 0$. Consequently, W_1 is invertible and we can define L_1 and L_2 as in (2.47).

Then, if we let $Q = W_1$ and define L_1 and L_2 as in (2.47), a calculation similar to that in the proof of Theorem 2.3 shows that

$$Q \geq (A - L_1 C_2)' Q (A - L_1 C_2) + (C_1 - L_2 C_2)'(C_1 - L_2 C_2),$$

so Q satisfies the Lyapunov equation for the closed-loop system. Also, $C_1 - L_2 C_2 = C_1 + V_2 C_2$, which is injective. Consequently, $(A - L_1 C_2, C_1 - L_2 C_2)$ is detectable. Therefore, by Proposition 2.5(b), $A - L_1 C_2$ must be stable, and L_1 is stabilizing. Moreover, using similar calculations as in the proof of Theorem 2.3, one can show that the asymptotic mean-squared error is bounded by

$$\begin{aligned} \sigma^2(L) &\leq \mathbf{Tr}\left((B - L_1 D_2)' Q (B - L_1 D_2) + (D_1 - L_2 D_2)'(D_1 - L_2 D_2)\right) \\ &\leq \mathbf{Tr}\left([B' \;\; D_2'] W [B' \;\; D_2']' + [D_1' \;\; D_2'] V [D_1' \;\; D_2']'\right), \end{aligned}$$

which proves part (a).

Conversely, consider any stabilizing matrices L_1 and L_2. Then, the closed-loop matrix $A - L_1C_2$ must be stable. Consequently, by Proposition 2.3, there exists a $Q \geq 0$ satisfying the Lyapunov equation for the closed-loop system,

$$Q = (A - LC_2)'Q(A - LC_2) + (C_1 - L_2C_2)'(C_1 - L_2C_2),$$

with

$$\sigma^2(L_1, L_2) = \mathbf{Tr}\left((B - L_1D_2)'Q(B - L_1D_2) + (D_1 - L_2D_2)'(D_1 - L_2D_2)\right).$$

If we define W and V as in (2.45) with

$$W_1 = Q, \quad W_2 = -QL, \quad W_3 = L'QL, \quad V_2 = -L_2, \quad \text{and} \quad V_3 = L_2'L_2,$$

then $W \geq 0$ and $V \geq 0$. Also, using similar calculations as before, one can show that W and V satisfy (2.49), which proves part (b).

2.8 Basic Principles of Quantization

Quantization is a vast field with an extensive literature. This section will review some basic facts from quantization theory and establish notation for later. All the material here is elementary and can be found in numerous textbooks such as [64] and in the excellent review and tutorial [75].

2.8.1 Definitions and Performance of Simple Fixed-Rate Quantizers

Mathematically, a *quantizer* is a mapping $Q(\cdot)$ from a continuous-valued quantity x, to a discrete quantity $Q(x)$. The quantizer output $Q(x)$ is restricted to take on one of a finite number, say M, possible values and is intended to represent a discrete approximation to x. The resolution of the quantizer is typically quoted in *bits* as $\log_2(M)$ bits. The average number of bits per sample of the quantizer is called its *rate*. The basic problem of quantization is to design the quantizer $Q(\cdot)$ such that $Q(x)$ is a good approximation of x, while keeping the rate small.

The design problem is typically posed probabilistically. The quantizer input x, often called the *source*, is modeled as a continuous random variable with a known p.d.f. Given a quantizer $Q(\cdot)$, one can then define the distortion D as some expected error. The simplest distortion is the mean-squared error

$$D = \mathbf{E}|x - Q(x)|^2. \tag{2.50}$$

The typical problem is: given a source x, find the quantizer $Q(\cdot)$ that minimizes the distortion D for a given number of bits per sample n.

The most common quantizer is an n-bit uniform scalar quantizer. It has $M = 2^n$ output levels, with neighboring pairs spaced by Δ. with *step size* Δ given by For a given number of bits n, the *step size* Δ must be selected based on the statistics of x. The curve labeled "uniform" in Figure 2.1 shows the best achievable quantizer accuracy using uniform quantization when x is a zero-mean Gaussian random variable with unit variance. The quantizer accuracy is plotted in terms its the reconstruction SNR in dB, with SNR defined as

$$\text{SNR} = \frac{\mathbf{E}|x|^2}{\mathbf{E}|x - Q(x)|^2}.$$

The reconstruction SNR is the inverse of the distortion relative to the quantizer input variance. The SNR is plotted in dB for a number of bits, or rate, ranging from 0 to 4. For each rate n, the step size Δ is optimized to find the minimum distortion, or equivalently, the maximum SNR. Asymptotically, for large rates R, the minimum distortion for uniform scalar quantization of a Gaussian random variable x is given by Bucklew and Gallagher [18] as

$$D_{\text{UQ}} \approx \sigma_X^2 K R 2^{-2R}, \tag{2.51}$$

where σ_X^2 is the variance of x and $K > 0$ is a constant that does not depend on the rate or the variance of x.

The simple uniform quantizer can be improved slightly by considering non-uniform spacing of the quantization levels. Given a p.d.f. on x and number of bits n, the optimal quantization points can be found by a well-known iterative procedure known as the Lloyd–Max algorithm [104, 110], which will be described in Section 2.8.3. The curve labeled "Lloyd–Max" in Figure 2.1 shows the achievable reconstruction SNR for non-uniform quantizers optimized with the Lloyd–Max procedure. It can be seen that Lloyd–Max quantization offers a small improvement over uniform quantization that increases with rate. For high rates R, a well-known result by Panter and Dite (see [75]) is that the minimum distortion using Lloyd–Max quantization for a Gaussian scalar random variable x is given by

$$D_{\text{LM}} \approx \sigma_x^2 \frac{\sqrt{3}\pi}{2} 2^{-2R}. \tag{2.52}$$

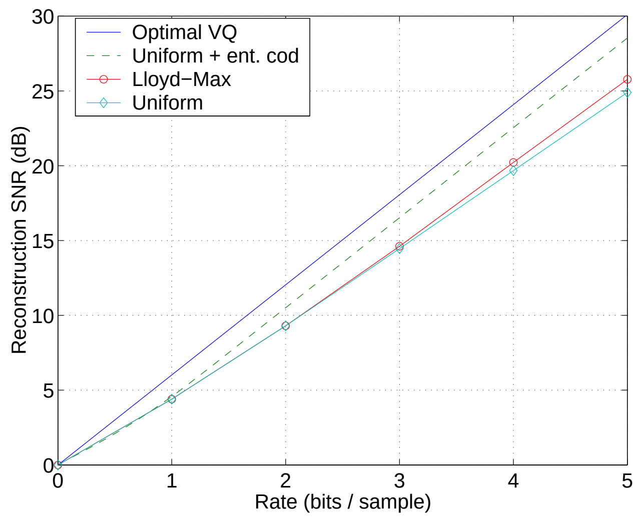

Figure 2.1. Quantization of a Gaussian source. Plotted is the reconstruction SNR vs. rate for: (a) theoretically optimal vector quantization (VQ), (b) bounded uniform scalar quantization with entropy coding, (c) Lloyd–Max optimal scalar quantization with no entropy coding, and (d) bounded uniform scalar quantization with no entropy coding.

Comparing D_{LM} with D_{UQ}, we see that simple uniform quantization has larger distortion asymptotically by a factor R. However, for small rates, the difference is not typically large. For example, for a rate of 4 bits per sample, Lloyd–Max quantization results in a 20.2 dB reconstruction SNR, only about 0.54 dB better than uniform quantization.

2.8.2 Entropy-Coded Quantization

Another way to improve on the uniform quantizer is through *entropy coding*. In general, the 2^n outputs of a uniform quantizer will not be equiprobable. For example, for a zero-mean Gaussian source x, the probability decreases with $|Q(x)|$. Consequently, one can combine the outputs of several quantized samples, and then apply a variable-length entropy code, such as a Huffman code, to reduce the average number of bits per sample. Suppose the quantizer has M outputs, $Q(x) \in \{q_1, \ldots, q_M\}$. If a large number of samples are entropy coded together, then the resulting rate can be reduced to the entropy of $Q(x)$ given by

$$R = -\sum_{i=1}^{M} \mathrm{Pr}(Q(x) = q_i) \log_2(\mathrm{Pr}(Q(x) = q_i).$$

For high rates R, a well-known analysis by Gish and Pierce [67] shows that the minimum distortion for uniform quantization with entropy coding for a Gaussian source is given by

$$D_{\mathrm{EC}} \approx \sigma_x^2 \frac{\pi e}{6} 2^{-2R}. \tag{2.53}$$

The corresponding reconstruction SNR is plotted in Figure 2.1; uniform quantization with entropy coding is shown to outperform Lloyd–Max quantization. The entropy-coded quantizer also has the appealing feature that it is relatively simple to implement and allows for arbitrary fractional rates by adjusting the step size appropriately. In particular, one can implement rates below 1 bit per sample, which is not possible with fixed-rate scalar quantization.

The above quantizers are *scalar* in that, given a large number of samples of x, each sample would be quantized individually. One could alternatively consider quantizing multiple samples together; this quantization is known as *vector quantization* (VQ). A fundamental result of information theory [41] is that the minimum achievable distortion with VQ for an i.i.d. Gaussian source is given by

$$D_{\mathrm{VQ}} = \sigma_x^2 2^{-2R}. \tag{2.54}$$

Comparing this distortion with (2.52) and (2.53), we see that, at high rates, the distortion with optimal VQ is better by a factor of $\sqrt{3}\pi/2$ or 4.35 dB over Lloyd–Max, and $\pi e/6$ or 1.53 dB over entropy-coded uniform scalar quantization. The minimum distortion D_{VQ} is called the *distortion–rate function* of a Gaussian random variable, since it describes the minimum achievable distortion as a function of the quantizer rate.

Unfortunately, optimal VQ is typically prohibitively difficult to implement. To vector quantize K samples with R bits per samples, requires 2^{RK} quantization values. The number of quantizer outputs thus grows exponentially with the number of samples K; this makes the implementation of an arbitrary VQ quantizer virtually impossible for even moderate values of K. Nevertheless, the minimum distortion (2.54) provides a useful theoretical limit to compare practical quantizers against.

2.8.3 Lloyd–Max Quantization

It is useful to briefly review the Lloyd–Max procedure for optimal quantizer design. Suppose x is a random variable with a continuous p.d.f., and let $Q(\cdot)$ be a quantizer with M possible output values, $Q(x) \in \{q_1, \ldots, q_M\}$. The general quantizer design problem is to select the M values $\{q_i\}$ and determine which values x get mapped to each q_i. One of the fundamental results of quantization is that a quantizer $Q(\cdot)$ locally minimizes the mean-squared error distortion metric (2.50) if and only if it satisfies the following conditions:

(a) For every x and every $i \in \{1, 2, \ldots, M\}$,

$$|Q(x) - x| \leq |q_i - x|; \quad \text{and}$$

(b) For every $i \in \{1, 2, \ldots, M\}$,

$$q_i = \mathbf{E}\left(x \mid Q(x) = q_i\right).$$

The conditions are due to Lloyd [104] and Max [110] and are relatively easy to understand: Property (a) states that the quantizer should map any vector x to the closest quantization point. Property (b) states that any quantizer output q_i should be the conditional expectation of x given that $Q(x) = q_i$.

The Lloyd–Max procedure finds the optimal values of q_i by alternately imposing the constraints (a) and (b) with the following algorithm:

1. Select a starting candidate set of q_i's.

2. For each i, find the set

$$S_i = \{x \,:\, |x - q_i| \leq |x - q_j|, \; \forall j\}.$$

3. Recompute a new set $\{q_i\}$ by

$$q_i = \mathbf{E}\left(x \mid x \in S_i\right).$$

4. Return to step 2.

Lloyd and Max show that, under suitable conditions, the algorithm converges to a set of values q_i satisfying the optimality conditions (a) and (b).

One consequence of property (b) that will be useful later is that, if the quantizer $Q(\cdot)$ is optimal,

$$Q(x) = \mathbf{E}\left(x \mid Q(x)\right).$$

This property of optimal quantizers will be used in Chapter 5.

Chapter 3

Estimation for Jump Linear Systems

3.1 Introduction

3.1.1 Jump Linear Estimation

A *jump linear system* is a time-varying linear state-space system driven by a discrete Markov chain. In this chapter, we consider state and output estimation for a discrete-time jump linear system of the form

$$\begin{aligned} x[k+1] &= A_{\theta[k]}x[k] + B_{\theta[k]}w[k], \\ z[k] &= C_{1,\theta[k]}x[k] + D_{1,\theta[k]}w[k], \\ y[k] &= C_{2,\theta[k]}x[k] + D_{2,\theta[k]}w[k]. \end{aligned} \tag{3.1}$$

The system is identical to the system (2.9) in the Kalman estimation problem, except that here the system matrices depend on a time-varying parameter $\theta[k]$. The parameter $\theta[k]$ is a discrete-time Markov chain with a finite number of possible states: $\theta[k] \in \{0, \dots, M-1\}$. Thus, at any time, each of the system matrices can take on one of M values. The system is called a "jump" linear system, since changes in the Markov state $\theta[k]$ result in discrete changes, or jumps, in the system dynamics. Due to the combination of the continuous state $x[k]$ and discrete state $\theta[k]$, jump linear systems are sometimes considered a type of *hybrid system.*

Jump linear systems were originally studied in control, where the Markov chain was used to model discrete dynamics that arise in faults, logic elements, or piecewise linear modeling of nonlinear systems. Jump linear models have proven to be extremely general and have found successful applications in a number of fields including fault detection in manufacturing, target tracking, guidance, finance and aerospace. See, for example, [4, 10, 14, 45, 143].

As discussed in Chapter 1, one of the key ideas in this dissertation is that jump linear systems can also be used to model time-varying channel conditions in communication problems. We will consider communication and robust coding applications in subsequent chapters. The purpose of this chapter is to develop the general theoretical and analytic framework for jump linear estimation.

The specific problem we consider for the system (3.1) is the estimation of the signal $z[k]$ from the signal $y[k]$ and the Markov state $\theta[k]$. Thus, $z[k]$ represents an unknown signal to be estimated, and $y[k]$ a sequence of observed samples. The estimation problem is identical to the Kalman estimation problem in Section 2.4, with the additional feature that the time variations in the system are described by a discrete Markov chain $\theta[k]$ known to the estimator.

Under the assumption that the Markov chain $\theta[k]$ is known, the estimator essentially sees a linear system with known time variations. Under suitable assumptions, the optimal estimator is given by a standard Kalman filter as described in Section 2.4. When applied to the jump linear system (3.1), the Kalman estimator has a simple recursive form: when $\theta[k] = i$,

$$\begin{aligned} \hat{x}[k+1|k] &= A_i\hat{x}[k|k-1] + L[k](y[k] - C_{i2}\hat{x}[k]) \\ \hat{z}[k|k-1] &= C_{i1}\hat{x}[k|k-1]. \end{aligned} \tag{3.2}$$

where $\hat{x}[k|k-1]$ and $\hat{z}[k|k-1]$ are the (strictly causal) estimates for $x[k]$ and $z[k]$, and $L[k]$ is a time-varying Kalman gain. In this way, for a given realization of the Markov sequence $\theta[k]$, and observation sequence $y[k]$, the Kalman estimator provides recursive equations for estimates of the state $x[k]$ and output $z[k]$. The Kalman filter also provides a recursive equation for the estimation error variance,

$$P[k|k-1] = \mathbf{E}\left((x[k] - \hat{x}[k|k-1])(x[k] - \hat{x}[k|k-1])'\right).$$

It would appear that the Kalman estimator provides the complete solution to the jump linear estimation problem. However, there are two issues that arise in applying the estimator to communication problems:

- Average estimation performance: The estimation error variance sequence, $P[k|k-1]$, provided by the Kalman filter, represents the error variance for a *given* realization of the Markov sequence $\theta[k]$. However, in many applications, what is relevant is not the performance of the estimator for a particular realization of the Markov sequence, but rather the *average* performance, averaged over all possible realizations of the Markov chain $\theta[k]$. For example, in communications problems, where the Markov chain is used to model time-variations in the channel quality, the performance of two encoder designs cannot be compared on a single channel realization. Instead, a more relevant metric is the achievable estimation error averaged over the possible channel realizations. The question is then how the average estimation error can be computed as a function of the system and Markov parameters.

- Estimation complexity: The optimal Kalman filter update involves updating a state estimation error variance matrix $P[k|k-1]$. This update is nonlinear and, in certain real-time applications, this update may be too computationally difficult. An important question is whether there are simpler, possibly suboptimal, estimators with acceptable performance.

3.1.2 Summary of the Main Results

It is difficult to compute the average performance of the optimal Kalman filter, due to the nonlinear update in the covariance error matrix. We thus consider a simpler, but suboptimal, estimator of the form: when $\theta[k] = i$,

$$\begin{aligned} \hat{x}[k+1] &= A_i\hat{x}[k] + L_i(y[k] - C_{i2}\hat{x}[k]) \\ \hat{z}[k] &= C_{i1}\hat{x}[k]. \end{aligned} \tag{3.3}$$

defined for some gain matrices L_i, $i = 1, \ldots, M$. The estimator (3.3) is identical to the Kalman estimator (3.2), except that the gain matrix $L[k]$ in the Kalman estimator is replaced by the gain matrix L_i. The difference between the estimators can be understood as follows: the Kalman gain matrix $L[k]$ depends, in principle, on the entire past sequence $\theta[j]$ from $j = 0$ to k. In the suboptimal estimator (3.3), the estimator gain L_i depends only on the current state of the Markov chain $\theta[k]$. We will call the suboptimal estimator (3.3) the *jump linear estimator*.

The gain matrices, L_i, $i = 0, \ldots, M-1$, for the jump linear estimator can be seen as design parameters. Our main results, Theorems 3.1 and 3.2, provide an LMI method for finding the gain matrices that minimize the asymptotic expected error $\mathbf{E}\|z[k]-\hat{z}[k]\|^2$. Theorem 3.1 provides an LMI optimization for the strictly causal estimator (3.3); Theorem 3.2 provides an LMI for a related causal estimator. The LMIs are a natural generalization of the LMIs in Section 2.7.2 and 2.7.3 which analyzed the LTI estimation problem.

The result can be interpreted in one of two ways: First, the jump linear estimator (3.3) is significantly simpler to implement than the optimal Kalman estimator (3.2). Once the gain matrices L_i are computed, the jump linear estimator is a simple time-varying system that avoids the state covariance update in the Kalman filter. In this regard, the LMI optimization provides a way of optimizing the performance of this simplified estimator.

Second, the LMI optimization is based on minimizing the *average* estimation performance, averaged over the realizations of the Markov chain $\theta[k]$. The estimation error of the optimized jump linear estimator, thus provides an upper bound on the minimum achievable estimation error of any estimator, including the optimal Kalman filter. Simulations in Chapter 4 will show that this LMI upper bound is often very close to the actual Kalman filter performance.

In addition to the main results, we will review, for completeness, some standard material in jump linear analysis including the stability and Lyapunov analysis of Costa and Fragoso [34], and the coupled Riccati equation solutions of Costa [29]. As a more theoretically minor, but useful result, we also provide are various simplifications for the case of i.i.d. Markov chains.

3.1.3 Previous Work

Jump linear systems have been studied in control since the 1960s, and there is now a considerable body of literature on the subject. The extent of the work in the area is a testament both to the utility of jump linear systems, and the progress made in their analysis. Costa, Fragoso and Marques have recently published a book [35] on the subject which provides a comprehensive overview of the field. While the book covers many of the historical details, it is useful here to briefly summarize the previous work and current state of the art in the area to place the results in this chapter in context.

Jump linear state-space models appeared at least as early as 1961 in the work of Krasovskii and Lidskii [95]. The optimal control of these systems was studied initially by Sworder [144], and later by Mariton and Bertrand [109]. These works showed that the optimal state-feedback control gain for a jump linear system could be described by a set of coupled Riccati differential equations that generalize the classic Riccati differential equations for the control of time-varying linear systems. However, the convergence of the coupled equations, and the related question of stabilizability of jump linear systems was not known. Early work by Costa and Fragoso [33] solved a special case of the problem with variations limited to the noise variance. Bitmead and Anderson [13], Li and Chizeck [101] and later Fang [55] gave somewhat complicated eigenvalue-like tests.

The complete solution to the problem came in 1995 where Costa [29] showed that the stabilizability of a jump linear system is equivalent to the existence of a solution to a set of coupled algebraic Riccati equations. If the system is stabilizable, the solution to the Riccati equations also provides the optimal state feedback control gains.

Costa's work [29] also considers the jump linear state estimation problem, which is mathematically the dual of the state-feedback control problem. For the estimation problem, Costa considers the suboptimal jump linear estimator (3.3) described above, and shows that the optimal gains L_i can also be solved by a set of coupled Riccati equations. The coupled equations for the filtering problem are the dual to the control equations. The filtering equations are also similar in form to the standard algebraic Riccati equation for LTI systems described in Section 2.4.3, except that, in the jump linear case, there are M equations, with one equation for each discrete state.

Since the work of Costa in [29], both the control and filtering coupled Riccati equations have been extensively studied. See, for example: Costa and Tueta [40] which proves an analogue of the separation principle for jump linear systems; Elliott, Dufour and Sworder [53, 48], which provide stochastic estimation interpretations of the filter Riccati equations; and Do Val, Geromel and Costa [46] which consider iterative solutions. For completeness, Costa's coupled filtering Riccati equations will be reviewed in Section 3.7 below.

A second major development in jump linear analysis was the work of Ait Rami and El Ghaoui [3], that showed that Costa's filtering and control coupled algebraic Riccati equations could be solved efficiently via a simple LMI optimization. The work not only solved the equations, but introduced LMI optimization into the study of jump linear systems.

However, while Ait Rami and El Ghaoui's LMI solution solved the coupled algebraic Riccati equations, their LMI described only the so-called *maximal solution*, which, for the state feedback control problem, provides the optimal state feedback gain. In 1997, Costa, do Val and Geromel [31] provided a more general LMI that could describe, not just the optimal solution, but the set of all state feedback control gains achieving a desired performance level. This more general LMI, sometimes called the *suboptimal* LMI, was much easier to generalize to other problem variants and was what really enabled much of the progress in LMI analysis of jump linear systems. At the present time, almost every imaginable extension, including extensions to continuous-time, H_∞ control, bounded noise disturbances, parametric uncertainties, and game theoretic interpretations, has been considered. See, for example, [26], [30], [32], [38], [39], [58], [106], [121], [137] as a small set of examples.

The general LMI solution in [31] and much of the following work considered only the jump linear control problem. The jump linear state estimation problem has received relatively less attention. Although the coupled Riccati equation solutions in [29] were provided for both the control and filtering problems, the LMI solution in [31] applies only to the control problem. Our main results, Theorems 3.1 and 3.2, can be seen as the estimation counterpart to the LMI state-feedback control LMI solution. In fact, we follow a derivation analogous to that in [31] that begins with the jump linear Lyapunov analysis of Costa and Fragoso in [34]. The final LMIs we obtain for the estimation problem are a natural dual to the control LMIs. Recently, De Souza and Fragoso [44] consider the continuous-time H_∞ estimation problem, as opposed to the stochastic estimation problem considered here.

One interesting aspect of the estimation problem that is usually not significant for state-feedback control, is delay. The direct analogue of the standard state-feedback control to the estimation problem results in a *strictly causal* estimator, where the estimate of the state $x[k]$ at time k depends on the observed data $y[j]$ up to and including the time $j = k - 1$. However, for our analysis, we will need to consider the *causal* estimation case also, where the estimator has access to the observation $y[j]$, for the current time $j = k$ also, not just for the times $j = 0$ to $k - 1$. The LMI solution of the causal jump linear estimation case is non-trivial, and it has not appeared before to the best of our knowledge.

A related line of research that should be mentioned at this point is the jump linear estimation problem when $\theta[k]$ is unknown to the estimator which was originally studied by Ackerson and Fu [1]. When $\theta[k]$ is unknown, the optimal estimator is to run a series of Kalman estimators in parallel, one for each possible $\theta[k]$ sequence. Since the number of sequences grows exponentially with the number of time samples, the exact optimal estimator is, in general, computationally impossible. Most of the research has focused on suboptimal algorithms. The best known is the IMM algorithm of Blom and Bar-Shalom [15], and a related procedure due to Doucet, Logothetis and Krishnamurthy [47]. The optimal linear

MMSE estimator was derived by Costa [28], and further studied in [37], [36] and [60]. It is important to recognize that the estimation problem with $\theta[k]$ known that is considered here is *not* a special case of the problem when $\theta[k]$ is unknown. There is no necessary relation between the minimum achievable estimation error between these two cases. The estimation problems for $\theta[k]$ unknown and known should thus be considered as separate problems, and one does not reduce to the other.

3.2 Mean Square Stability and Detectability

Before considering the estimation problem, we first need to review some standard properties and definitions for jump linear systems. This section, and the following two sections on jump linear Lyapunov analysis, will cover standard results from [34]. The material is also covered in [29] and the book [35].

We begin with the issue of stability. Consider a jump linear system of the form,

$$\begin{aligned} x[k+1] &= A_{\theta[k]}x[k] + B_{\theta[k]}w[k], \\ z[k] &= C_{\theta[k]}x[k] + D_{\theta[k]}w[k] \end{aligned} \tag{3.4}$$

where $\theta[k]$ is some Markov chain. For this system, we can define stability as follows.

Definition 3.1 *Consider the jump linear system in (3.4) driven by a Markov chain* $\theta[k]$*. We will say the system is* mean square (MS) stable *if: for any initial condition,* $(x[0], \theta[0])$*,*

$$\lim_{k\to\infty} \mathbf{E}\|x[k]\|^2 = 0$$

whenever $w[k] = 0$*. The expectation here is with respect to the realizations of the Markov chain* $\theta[k]$*. We will say that a set of matrices* A_i*,* $i = 0, \ldots, M-1$*, are* MS stable with respect to the Markov chain $\theta[k]$ *if the corresponding jump linear system (3.4) is MS stable.*

Comparing Definition 3.1 with Proposition 2.1, we see that MS stability, as it is defined here, is a natural extension of the standard definition of stability for LTI systems. Indeed, in the special case when the jump linear system (3.4) has only $M = 1$ state, the jump linear system reduces to a time-invariant system and MS stability and stability in the sense of Definition 2.1 are equivalent.

However, there are two important differences between stability for jump linear systems and LTI systems. First, stability for a jump linear system is a probabilistic concept and depends, not only on the A_i matrices, but also on the statistics of the Markov chain, $\theta[k]$. Second, for jump linear systems, there are no simple eigenvalue tests on the matrices A_i that is necessary and sufficient for stability. For example, it is *not* in general true that if all the matrices A_i have eigenvalues in the unit disc, the jump linear system will be stable.

In addition to stability, we will need an analogue of detectability. Recall that for an LTI system, a matrix pair (A, C) is detectable if there exists a matrix L such that $A - LC$ is stable. The natural extension of this definition for jump linear systems is provided by the following definition.

Definition 3.2 *Consider the jump linear system in (3.4) driven by a Markov chain $\theta[k]$. The matrices (A_i, C_{i2}) will be called* mean square (MS) detectable *with respect to $\theta[k]$ if there exist matrices L_i such that the set of matrices $A_i - L_iC_{i2}$ is MS stable. Any set of gain matrices L_i resulting in $A_i - L_iC_{i2}$ being MS stable will be called* MS stabilizing.

Again, note that MS detectability is implicitly a function of the statistics on the Markov chain as well as the system matrices. We will see that MS detectability plays a similar role in estimation for jump linear systems as detectability does for LTI systems.

Definition 3.1 use a mean-square convergence criteria. One could also consider other forms of convergence, such as exponential or almost sure convergence. For the most part, for finite state linear systems with finite Markov chains, these conditions are often equivalent. The interested reader is referred to Costa and do Val [27] for related concepts of stability.

3.3 Coupled Lyapunov Equations

We saw that for LTI systems, stability was closely related to the existence of solutions to so-called Lyapunov equations. For jump linear systems, a similar set of equations exists, except that the single Lyapunov equation is replaced by a system of M equations called *coupled Lyapunov equations*. These equations are well-known, and here we will follow the presentation in Costa [34]. We will restrict our attention to stationary Markov processes as described in the following assumption.

Assumption 3.1 *The random sequence $\theta[k]$ is a stationary Markov chain, with transition* *probabilities,*

$$p_{ij} = \mathbf{Pr}\left(\theta[k+1] = j \mid \theta[k] = i\right).$$

We assume that the Markov chain is aperiodic and irreducible so that there is a unique stationary distribution,

$$q_i = \mathbf{Pr}\left(\theta[k] = i\right),$$

satisfying the equations,

$$q_j = \sum_{i=0}^{M-1} p_{ij} q_i, \quad j = 0, \ldots, M-1.$$

Under this assumption, the Lyapunov equations for the system (3.4) can be described as follows.

Proposition 3.1 *Consider the jump linear system in (3.4) driven by a Markov chain* $\theta[k]$ *satisfying Assumption 3.1. Let* $S_j \geq 0$, $j = 0, \ldots, M-1$, *be any set of positive semidefinite matrices.*

(a) If the matrices A_j *are MS stable with respect to* $\theta[k]$, *there exists a unique set of matrices* P_j *satisfying the* coupled Lyapunov equations,

$$P_j = A_j \overline{P}_j A_j' + S_j, \tag{3.5}$$

for all $j = 0, \ldots, M-1$, *where*

$$\overline{P}_j = \frac{1}{q_j} \sum_{i=0}^{M-1} p_{ij} q_i P_i$$

The matrices P_j *satisfy* $P_j \geq S_j$. *In particular,* $P_j \geq 0$.

(b) Conversely, if $S_j > 0$ *for all* j, *and there exist matrices* $P_j \geq 0$ *satisfying (3.5) then the matrices* A_j *are MS stable with respect to* $\theta[k]$.

The coupled Lyapunov equations (3.5) are similar in form to the standard Lyapunov equation (2.6) for an LTI system. The main difference is that, for the jump linear system, there are M equations (3.5) and M unknowns P_j. Moreover, the equations are *coupled* since each equation depends on all matrices P_i. Observe that, for a fixed set of transition probabilities, p_{ij}, the equations are linear in the matrices P_j.

Similar to (2.8), it also useful to consider a dual Lyapunov equation, as described in the following result.

Proposition 3.2 *Consider the jump linear system in (3.4) driven by a Markov chain $\theta[k]$ satisfying Assumption 3.1. Let $S_i \geq 0$, $i = 0, \ldots, M-1$, be any set of positive semidefinite matrices.*

(a) If the matrices A_i are MS stable with respect to $\theta[k]$, there exists a unique set of matrices Q_i satisfying the dual coupled Lyapunov equations,

$$Q_i = A_i \overline{Q}_i A_i' + S_i \tag{3.6}$$

for all $i = 0, \ldots, M-1$, where $\overline{Q}_i$ is defined by,

$$\overline{Q}_i = \sum_{j=0}^{M-1} p_{ij} Q_j$$

The matrices Q_i satisfy $Q_i \geq S_i$. In particular, $Q_i \geq 0$.

(b) Conversely, if $S_i > 0$ for all i, and there exist matrices $Q_i \geq 0$ satisfying (3.6), then the matrices A_i are MS stable with respect to $\theta[k]$.

3.4 Lyapunov Analysis

Consider again the jump linear system (3.4) and suppose that the input $w[k]$ is a zero-mean, white random process with unit variance. As in the LTI case, the Lyapunov equations can be used to compute the steady-state state and output variances. Specifically, as in Section 2.3, define the expectations of the state and output,

$$\hat{x}[k] = \mathbf{E}x[k], \quad \hat{z}[k] = \mathbf{E}z[k].$$

Here, the expectations are over both the noise $w[k]$ *and* the discrete state sequence $\theta[k]$. Let $P[k]$ denote corresponding state covariance matrix,

$$P[k] = \mathbf{E}(x[k] - \hat{x}[k])(x[k] - \hat{x}[k])', \tag{3.7}$$

and let $\sigma^2[k]$ denote the output variance,

$$\sigma^2[k] = \mathbf{E}\|z[k] - \hat{z}[k]\|^2. \tag{3.8}$$

As before, we will be interested in the asymptotic quantities,

$$\lim_{k\to\infty} P[k], \quad \lim_{k\to\infty} \sigma^2[k]. \tag{3.9}$$

The following results shows that these asymptotic values can be computed from the coupled Lyapunov equations.

Proposition 3.3 *Consider the jump linear system in (3.4) driven by a Markov chain $\theta[k]$ satisfying Assumption 3.1. Suppose that the input $w[k]$ is a zero-mean, white random process with unit variance. Let $P[k]$ and $\sigma^2[k]$ be the state covariance matrix and output variance defined in (3.7) and (3.8), respectively. Then,*

(a) If the system is MS stable, there exist matrices $P_j \geq 0$, $j = 0, \ldots, M-1$ satisfying,

$$P_j = A_j \overline{P}_j A_j' + B_j B_j', \tag{3.10}$$

for all $j = 0, \ldots, M-1$, where

$$\overline{P}_j = \frac{1}{q_j} \sum_{i=0}^{M-1} p_{ij} q_i P_i \tag{3.11}$$

The asymptotic state and output variances are given by,

$$\begin{aligned} \lim_{k\to\infty} P[k] &= \sum_{j=0}^{M-1} q_j P_j, \\ \lim_{k\to\infty} \sigma^2[k] &= \sum_{j=0}^{M-1} q_j \mathbf{Tr}\left(C_j \overline{P}_j C_j' + D_j D_j'\right). \end{aligned}$$

(b) Conversely, suppose $B_j B_j' > 0$ for all j, and that there exist matrices $P_j \geq 0$ satisfying

$$P_j \geq A_j \overline{P}_j A_j' + B_j B_j', \tag{3.12}$$

where $\overline{P}_j$ is defined in (3.11). Then, the system is MS stable and the asymptotic variances are bounded by,

$$\begin{aligned} \lim_{k\to\infty} P[k] &\leq \sum_{j=0}^{M-1} q_j P_j, \\ \lim_{k\to\infty} \sigma^2[k] &\leq \sum_{j=0}^{M-1} q_j \mathbf{Tr}\left(C_j \overline{P}_j C_j' + D_j D_j'\right). \end{aligned}$$

Part (a) of Proposition 3.3 is a natural extension of Proposition 2.2 for jump linear systems: Proposition 2.2 showed that the asymptotic variance of an LTI system can be computed from a Lyapunov equation. The result here shows that, for a jump linear system, the asymptotic state and output variances can be computed from M coupled Lyapunov equations.

Similarly, part (b) of Proposition 3.3 is an extension of the LTI result in Proposition 2.5(a), and shows that the output variance of a jump linear system state can be bounded by solving M coupled Lyapunov inequalities. This bound can be minimized with an LMI as follows: Combining parts (a) and (b) shows that the asymptotic output variance is given by,

$$\lim_{k\to\infty} \sigma^2[k] = \min_{P_j} \sum_{j=0}^{M-1} q_j \mathbf{Tr}\left(C_j \overline{P}_j C_j' + D_j D_j'\right). \tag{3.13}$$

where the minimization is over matrices $P_j \geq 0$ satisfying (3.12). For a fixed set of transition probabilities p_{ij}, the matrices P_j appear linearly in the objective (3.13) and constraint (3.12) of this optimization. Consequently, the optimization can be performed as an LMI, thereby providing a computationally simple way of computing the asymptotic output variance.

Similar to Propositions 2.3 and 2.5(b), we can also compute the asymptotic output variance by a dual Lyapunov-type equation as shown in the following result.

Proposition 3.4 *Consider the jump linear system in (3.4) driven by a Markov chain $\theta[k]$ satisfying Assumption 3.1. Suppose that the input $w[k]$ is a zero-mean, white random process with unit variance. Let $\sigma^2[k]$ be the output variance defined in (3.8). Then,*

(a) If the system is MS stable, there exist matrices $Q_j \geq 0$, $j = 0, \ldots, M-1$ satisfying,

$$Q_i = A_i' \overline{Q}_i A_i + C_i' C_i, \quad i = 0, \ldots, M-1 \tag{3.14}$$

where

$$\overline{Q}_i = \sum_{j=0}^{M-1} p_{ij} Q_j. \tag{3.15}$$

The asymptotic output variance is given by,

$$\lim_{k\to\infty} \sigma^2[k] = \sum_{j=0}^{M-1} q_i \mathbf{Tr}\left(B_i' \overline{Q}_i B_i + D_i' D_i\right).$$

(b) Conversely, suppose $C_i'C_i > 0$ for all i, and there exist matrices $Q_j \geq 0$ satisfying

$$Q_i \geq A_i'\overline{Q}_iA_i + C_i'C_i, \quad i = 0, \ldots, M-1, \tag{3.16}$$

where $\overline{Q}_i$ is defined as in (3.15). Then the system is MS stable and

$$\lim_{k\to\infty} \sigma^2[k] \leq \sum_{j=0}^{M-1} q_i \mathbf{Tr}\left(B_i'\overline{Q}_iB_i + D_i'D_i\right).$$

3.5 Jump Linear Estimation

We can now turn to the estimation problem. Consider the jump linear system (3.1) where the input $w[k]$ is zero-mean, white noise with unit variance. As discussed before, we are interested in the estimation of the state, $x[k]$, and output, $z[k]$, from the signal, $y[k]$, and Markov sequence, $\theta[k]$. We will denote these estimates by

$$\begin{aligned} \hat{x}[k|j] &= \mathbf{E}\left(\ x[k] \mid y[0:j], \theta[0:k]\ \right), \\ \hat{z}[k|j] &= \mathbf{E}\left(\ z[k] \mid y[0:j], \theta[0:k]\ \right). \end{aligned} \tag{3.17}$$

In this notation, $\hat{x}[k|j]$ and $\hat{z}[k|j]$ are the MMSE estimates of $x[k]$ and $z[k]$ given the signal y from samples 0 to j, and the sequence θ from samples 0 to k.

As discussed earlier, given the sequence $\theta[k]$, the system (3.1) reduces to a linear time-varying system with known system matrices. Consequently, if $w[k]$ is Gaussian, the MMSE estimates (3.17) are given by the time-varying Kalman filter in Section 2.4. If $w[k]$ is not Gaussian, the Kalman filter provides the linear MMSE estimate. In either case, for the jump linear system (3.1), the Kalman filter will take the form that, when $\theta[k] = i$,

$$\begin{aligned} \hat{x}[k+1|k] &= A_i\hat{x}[k|k-1] + L_1[k](y[k] - C_{i2}\hat{x}[k|k-1]) \\ \hat{z}[k|k-1] &= C_{i1}\hat{x}[k|k-1], \end{aligned} \tag{3.18}$$

where $L_1[k]$ is a time-varying Kalman matrix given by the Riccati equation recursions (2.12). The matrix $L_1[k]$ will depend on the specific realization of the sequence $\theta[k]$.

Unfortunately, the steady-state performance of the Kalman estimator (3.18) is somewhat difficult to analyze. The chief difficulty is that the Riccati equation update for $P[k|k-1]$ in (2.12) is a complex nonlinear recursion. As discussed in the introduction of this chapter, we thus consider a simpler, suboptimal, estimator of the form: when $\theta[k] = i$,

$$\begin{aligned} \hat{x}[k+1] &= A_i\hat{x}[k] + L_i(y[k] - C_{i2}\hat{x}[k]) \\ \hat{z}[k] &= C_{i1}\hat{x}[k], \end{aligned} \tag{3.19}$$

defined for some matrices L_i, $i = 0, \ldots, M-1$. We call the estimator (3.19) a *jump linear estimator*. The jump linear estimator is identical to the Kalman filter (3.18), except that the gain matrix $L_1[k]$ is replaced by the M fixed matrices L_i. Consequently, the gain matrix for the jump linear estimator is restricted to depend on only the current value of $\theta[k]$. In contrast, the optimal gain matrix $L_1[k]$ in (3.18) is a result of the Riccati equation recursion and consequently depends on all the values of $\theta[j]$ from times $j = 0$ to k.

3.6 LMI Analysis

3.6.1 Strictly Causal Estimation

The estimator (3.19) with fixed gain matrices can be analyzed relatively easily using LMIs. Let $L = (L_0, \ldots, L_{M-1})$ denote a set of gain matrices for the estimator, and, define the asymptotic error variance,

$$\sigma^2(L) = \lim_{k\to\infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2, \tag{3.20}$$

where the dependence on L is through the estimate $\hat{z}[k]$. The LMI analysis will attempt to find the set of matrices L that minimizes this asymptotic error,

$$\min_L \sigma^2(L). \tag{3.21}$$

That is, we regard the matrices L as a set of design parameters for the estimator (3.19), and attempt to optimize the estimator performance.

We can regard the optimization (3.21) in one of two ways. On the one hand, we can regard the optimization as a design method to maximize the performance of the estimator (3.19). The resulting estimator is useful in its own right. Although the estimator (3.19) is not optimal, it is significantly simpler to implement than the optimal Kalman filter (3.18). In each iteration of the Kalman filter, one must perform the Riccati recursion to update the state variance and compute $L_1[k]$. In the simplified estimator (3.19), the gain matrices L_i can be pre-computed and do not require any real-time computations. If gain matrices L_i can be found with suitable performance, one can realize significant computational savings.

Alternatively, we can use the minimization (3.21) as an analytic tool. Specifically, since the Kalman filter is optimal, the performance of any estimator of the form (3.19) provides an *upper bound* on the Kalman filter performance. Performing the minimization (3.21) minimizes that upper bound.

In either interpretation of the optimization, we will restrict our attention to MS stabilizing gain matrices L as defined earlier in Definition 3.2. The motivation for restricting to such stabilizing matrices is provided by the following simple proposition which is the analogue of Proposition 2.6.

Proposition 3.5 *Consider the jump linear system in (3.1) driven by a Markov chain $\theta[k]$ satisfying Assumption 3.1. Also, given a set of gain matrices,*

$$L = (L_0, \ldots, L_{M-1}),$$

let $\hat{x}[k]$ be the state estimate from the estimator in (3.19). Then, the gain matrices L are MS stabilizing if and only if: for any initial condition $(x[0], \hat{x}[0], \theta[0])$,

$$\lim_{k \to \infty} \mathbf{E}\|x[k] - \hat{x}[k]\|^2 = 0 \tag{3.22}$$

when $w[k] = 0$ for all $k = 0$. The expectation is with respect to the realizations of the Markov chains $\theta[k]$.

Proof: When $w[k] = 0$, we can subtract the state update equations in (3.4) and (3.19), to obtain the recursion that, when $\theta[k] = i$,

$$x[k+1] - \hat{x}[k+1] = (A_i - L_i C_{i2})(x[k] - \hat{x}[k]).$$

Now by Definition 3.1, the limit in (3.22) holds if and only if the set of matrices $A_i - L_i C_{i2}$ are MS stable. But, by Definition 3.2, matrices $A_i - L_i C_{i2}$ being MS stable is precisely the definition of the matrices L_i being MS stabilizing. □

We can now state the main result.

Theorem 3.1 *Consider the jump linear system in (3.1) driven by a Markov chain $\theta[k]$ satisfying Assumption 3.1. Suppose $w[k]$ is zero-mean, white noise with unit variance independent of $\theta[k]$. Given a set of gain matrices*

$$L = (L_0, \ldots, L_{M-1}),$$

let $\hat{z}[k]$ be the estimator output in (3.19), and let $\sigma^2(L)$ be asymptotic mean-squared output estimation error (3.20).

(a) Suppose that C_{i1} is injective for all i, and suppose that there exist matrices W_i, $i = 0, \dots, M-1$, partitioned as

$$W_i = \begin{bmatrix} W_{i1} & W_{i2} \\ W_{i2}' & W_{i3} \end{bmatrix}, \tag{3.23}$$

satisfying

$$\begin{aligned} W_{i1} &\geq [A_i' \; C_{i2}']\overline{W}_i[A_i' \; C_{i2}']' + C_{i1}'C_{i1}, \\ \overline{W}_i &\geq 0 \end{aligned} \tag{3.24}$$

where $\overline{W}_i$ is defined by

$$\overline{W}_i = \begin{bmatrix} \overline{W}_{i1} & W_{i2} \\ W_{i2}' & W_{i3} \end{bmatrix}, \tag{3.25}$$

and

$$\overline{W}_{i1} = \sum_{j=0}^{M-1} p_{ij} W_{j1}. \tag{3.26}$$

Then, $\overline{W}_{i1} > 0$ for all i. Also, if we define

$$L_i = -\overline{W}_{i1}^{-1} W_{i2}, \tag{3.27}$$

the set of matrices $L = (L_0, \dots, L_{M-1})$ is MS stabilizing and the asymptotic mean-squared error is bounded by

$$\sigma^2(L) \leq \sum_{i=0}^{M-1} q_i \mathbf{Tr}\left([B_i' \; D_{i2}']\overline{W}_i[B_i' \; D_{i2}']') + D_{i1}'D_{i1}\right). \tag{3.28}$$

(b) Conversely, for any set of MS stabilizing gain matrices L, there must exist W_i satisfying (3.23) and (3.24) with

$$\sum_{i=0}^{M-1} q_i \mathbf{Tr}\left([B_i' \; D_{i2}']\overline{W}_i[B_i' \; D_{i2}']') + D_{i1}'D_{i1}\right) \leq \sigma^2(L). \tag{3.29}$$

The result can be seen as an extension of Theorem 2.3 to jump linear systems. In the case of time-invariant systems, i.e. there is only $M = 1$ state, the jump linear result in Theorem 3.1 reduces to Theorem 2.3.

Combining parts (a) and (b) of Theorem 3.1, we see that the minimum asymptotic estimation error is given by,

$$\min_L \sigma^2(L) = \min_{W_i} \sum_{i=0}^{M-1} q_i \mathbf{Tr}\left([B_i' \; D_{i2}']W_i[B_i' \; D_{i2}']') + D_{i1}'D_{i1}\right), \tag{3.30}$$

where the first minimization is over MS stabilizing gain matrices L, and the second minimization is over matrices W_i satisfying (3.23) and (3.24). For a fixed set of transition probabilities, p_{ij}, the objective function (3.30) and constraint (3.24) are linear in the variables W_i. Consequently, the optimization can be solved as an LMI, thus providing a simple way to optimize the jump linear estimator.

3.6.2 Causal Estimation

The estimator in (3.19) is strictly causal in that the estimate $\hat{z}[k]$ depends on the samples $y[j]$ from $j = 0$ to $k - 1$. We can also consider a causal estimator, where we permit the estimator to depend on the samples $y[j]$ for $j = k$ as well. Similar to the time-invariant causal estimator in (2.43), we will consider a jump linear causal estimator of the form: when $\theta[k] = i$,

$$\begin{aligned} \hat{x}[k+1] &= A_i\hat{x}[k] + L_{i1}(y[k] - C_{i2}\hat{x}[k]) \\ \hat{z}[k] &= C_{i1}\hat{x}[k] + L_{i2}(y[k] - C_{i2}\hat{x}[k]). \end{aligned} \tag{3.31}$$

The estimator is defined for a set of gain matrices,

$$\begin{aligned} L_1 &= (L_{0,1}, \ldots, L_{M-1,1}), \\ L_2 &= (L_{0,2}, \ldots, L_{M-1,2}). \end{aligned} \tag{3.32}$$

Given gain matrices (L_1, L_2), we can define the asymptotic estimation error,

$$\sigma^2(L_1, L_2) = \lim_{k\to\infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2, \tag{3.33}$$

where $\hat{z}[k]$ is the estimate in (3.31). The following result is an extension of Theorem 2.4 to jump linear systems, and provides an LMI method for minimizing the asymptotic estimation error for the causal estimator.

Theorem 3.2 *Consider the jump linear system in (3.4) driven by a Markov chain $\theta[k]$ satisfying Assumption 3.1. Suppose $w[k]$ is zero-mean, white noise with unit variance independent of $\theta[k]$. Given sets of gain matrices (L_1, L_2) as in (3.32), let $\hat{z}[k]$ be the estimator output in (3.31), and let $\sigma^2(L_1, L_2)$ be asymptotic mean-squared output estimation error (3.33).*

(a) Suppose that $[C_{i1}' \; C_{i2}']$ is onto for all i, and suppose that there exist matrices W_i and

V_i, $i = 0, \ldots, M-1$, partitioned as

$$W_i = \begin{bmatrix} W_{i1} & W_{i2} \\ W_{i2}' & W_{i3} \end{bmatrix}, \quad V_i = \begin{bmatrix} I & V_{i2} \\ V_{i2}' & V_{i3} \end{bmatrix}, \tag{3.34}$$

satisfying

$$\begin{aligned} W_{i1} &\geq [A_i' \ C_{i2}']\overline{W}_i[A_i' \ C_{i2}']' + [C_{i1}'C_{i2}']V_i[\ C_{i1}' \ C_{i2}']' \\ \overline{W}_i &\geq 0 \\ V_i &\geq 0 \end{aligned} \tag{3.35}$$

where $\overline{W}_i$ is defined in (3.25). Then, $\overline{W}_{i1} > 0$ for all i. Also, if we define

$$L_{i1} = -\overline{W}_{i1}^{-1}W_{i2}, \quad L_{i2} = -V_{i2}, \tag{3.36}$$

the set of matrices (L_1, L_2) is MS stabilizing and the asymptotic mean-squared error is bounded by

$$\begin{aligned} \sigma^2(L_1, L_2) &\leq \sum_{i=0}^{M-1} q_i \mathbf{Tr}([B_i' \ D_{i2}']\overline{W}_i[B_i' \ D_{i2}']' \\ &+ [D_{i1}' \ D_{i2}']V_i[D_{i1}' \ D_{i2}']'). \end{aligned} \tag{3.37}$$

(b) *Conversely, for any set of MS stabilizing gain matrices (L_1, L_2), there exist matrices W_i and V_i satisfying (3.35) and*

$$\sum_{i=0}^{M-1} q_i \mathbf{Tr}([B_i' \ D_{i2}']\overline{W}_i[B_i' \ D_{i2}']' + [D_{i1}' \ D_{i2}']V_i[D_{i1}' \ D_{i2}']') \leq \sigma^2(L_1, L_2) \tag{3.38}$$

3.7 Coupled Riccati Equations

The previous section has shown that the jump linear estimation problem can be solved as an LMI. However, while the LMI solution is computationally attractive, for analysis, it is sometimes useful to have an analogue to the algebraic Riccati equation that was discussed earlier in Section 2.4.3 for LTI systems. As discussed in Section 3.1.3, one of the major results in jump linear analysis is that the filtering problem can be solved as a set of nonlinear *coupled* Riccati equations. The coupled Riccati solution was originally proved by Costa in [29]. For completeness, we will restate and prove the result here, in a slightly modified form, for clarity and ease of use in our later results and proofs.

Theorem 3.3 *Consider the jump linear system in (3.4) driven by a Markov chain* $\theta[k]$ *satisfying Assumption 3.1. Suppose that the matrix* $[B_i' \; D_{i2}']$ *is injective for all* i*. Then,*

(a) The jump linear system (3.4) is MS detectable if and only if there exist matrices $P_j \geq 0$*,* $j = 0, \ldots, M-1$*, satisfying*

$$P_j = A_j \overline{P}_j A_j' - E_{j1} G_j E_{j1}' + B_j B_j' \tag{3.39}$$

where $\overline{P}_j$ *is defined in (3.11) and*

$$\begin{aligned} E_{j1} &= A_j \overline{P}_j C_{j2}' + B_j D_{j2} \\ G_j &= (C_{j2} \overline{P}_j C_{j2}' + D_{j2} D_{j2})^{-1}. \end{aligned} \tag{3.40}$$

(b) Now, suppose that (3.4) is MS detectable and $w[k]$ *is zero-mean, white noise with unit variance independent of* $\theta[k]$*. Given a set of gain matrices*

$$L = (L_0, \ldots, L_{M-1})$$

let $\hat{z}[k]$ *be the estimator output in (3.19), and let* $\sigma^2(L)$ *be asymptotic mean-squared output estimation error (3.20). Then, there exists a set of* $P_i \geq 0$*, satisfying (3.39), such that the minimum estimation error is given by,*

$$\min_L \sigma^2(L) = \sum_{j=0}^{M-1} q_j \mathbf{Tr}(C_{j1} \overline{P}_j C_{j1}' + D_{j1} D_{j1}'),$$

where the minimization is over the set of MS stabilizing gain matrices L*. Moreover, one set of minimizing gain matrices is given by*

$$L_j = E_{j1} G_j$$

where E_{j1} *and* G_j *are given in (3.40).*

The equations (3.39) are called the *coupled algebraic Riccati equations*, since they represent a system of M equations, each similar in form to the standard algebraic Riccati equation (2.17). The equations (3.39) are called coupled since each equation depends on all matrices P_j.

The coupled Riccati equations (3.39) are, in general, non-linear due to the $E_{j1} G_j E_{j1}$ terms. Consequently, they are difficult to solve, and it is easier to find the optimal gain matrices L_j with the LMI method in Section 3.6 rather than using Theorem 3.3. Theorem 3.3 will thus serve more as an analytic tool than a computational one.

3.8 Independent Markov Chains

Suppose the Markov chain, $\theta[k]$, is an independent identically distributed (i.i.d.) process. In this case, the transition probabilities satisfy

$$p_{ij} = q_j$$

for all i and j. For such processes the LMI equations in the previous section significantly simplify. For example, consider first the Lyapunov result in Propositions 3.3 and 3.4. For the case of an i.i.d. Markov chain, the result simplifies to the following.

Proposition 3.6 *Consider the jump linear system in (3.4) driven by a Markov chain* $\theta[k]$ *satisfying Assumption 3.1. Suppose the Markov chain is i.i.d. so that* $p_{ij} = q_j$ *for all* i *and* j*. Also suppose* $B_i B_i' > 0$ *and* $C_i' C_i > 0$ *for all* i*. Let* $\sigma^2[k]$ *be output variance defined in (3.8). Then, the following are equivalent,*

(a) The system is MS stable and

$$\lim_{k\to\infty} \sigma^2[k] \leq \gamma.$$

(b) There exists a $P \geq 0$ *satisfying*

$$P \geq \sum_{j=0}^{M-1} q_j \left[A_j P A_j' + B_j B_j' \right], \tag{3.41}$$

and

$$\mathbf{Tr}\left(C_j P C_j' + D_j D_j' \right) \leq \gamma. \tag{3.42}$$

(c) There exists a $Q \geq 0$ *satisfying*

$$Q \geq \sum_{j=0}^{M-1} q_j \left[A_j' Q A_j + C_j' C_j \right], \tag{3.43}$$

and

$$\mathbf{Tr}\left(B_j' Q B_j + D_j' D_j \right) \leq \gamma.$$

Proof: We prove the equivalence of (a) and (b). The proof for the equivalence of (a) and (c) is similar.

So suppose (a) is true. Then, Proposition 3.3(a) shows that there exist matrices $P_i \geq 0$,

$$P_j = A_j \overline{P}_j A_j' + B_j B_j' \tag{3.44}$$

where $\overline{P}_j$ is defined in (3.11) and

$$\sum_{j=0}^{M-1} q_j \mathbf{Tr}\left(C_j \overline{P}_j C_j' + D_j D_j'\right) \leq \gamma. \tag{3.45}$$

Now, substituting $p_{ij} = q_j$ into (3.11), we obtain

$$\begin{aligned} \overline{P}_j &= \frac{1}{q_j} \sum_{i=0}^{M-1} p_{ij} q_i P_i = \frac{1}{q_j} \sum_{i=0}^{M-1} q_j q_i P_i \\ &= \sum_{i=0}^{M-1} q_i P_i. \end{aligned}$$

Therefore, if we let $P = \sum_i q_i P_i$, we see that $\overline{P}_j = P$ for all j. Using this fact along with (3.44),

$$\begin{aligned} P &= \sum_{j=0}^{M-1} q_j P_j \\ &= \sum_{j=0}^{M-1} q_j \left[A_j P A_j' + B_j B_j'\right], \end{aligned}$$

which shows that P satisfies (3.41). Also substituting $\overline{P}_j = P$ into (3.45) proves that P satisfies (3.42). Therefore, (a) $\Rightarrow$ (b).

To show that (b) $\Rightarrow$ (a), suppose there exists a matrix $P \geq 0$ satisfying (3.41) and (3.42) in part (b). Set

$$P_j = q_j \left[A_j P A_j' + B_j B_j'\right]. \tag{3.46}$$

Combining this definition with (3.11), (3.41) and the fact that $p_{ij} = q_j$,

$$\begin{aligned} \overline{P}_j &= \frac{1}{q_j} \sum_{i=0}^{M-1} p_{ij} q_i P_i = \frac{1}{q_j} \sum_{i=0}^{M-1} q_j q_i P_i \\ &= \sum_{i=0}^{M-1} q_i P_i = \sum_{i=0}^{M-1} q_i \left[A_j P A_j' + B_j B_j'\right] \geq P. \end{aligned}$$

Therefore $\overline{P}_j \geq P$ for all P. Substituting this into (3.46) shows P_j satisfy (3.12). Therefore, using Proposition 3.3(a) along with (3.42) and the fact that $\overline{P}_j \leq P$ shows that

$$\begin{aligned}
&\sum_{j=0}^{M-1} q_j \mathbf{Tr}\left(C_j P C_j' + D_j D_j'\right) \\
&\leq \sum_{j=0}^{M-1} q_j \mathbf{Tr}\left(C_j P_j C_j' + D_j D_j'\right) \\
&= \lim_{k\to\infty} \sigma^2[k] \leq \gamma.
\end{aligned}$$

which proves part (b). □

The LMI in the above proposition is significantly simpler to solve than the LMI for a general jump linear system. For example, in part (b), the asymptotic output variance can be computed by minimization over a single matrix $P \geq 0$ with a single constraint (3.41). In contrast, in Proposition 3.3 for general jump linear systems, the minimization involves M matrices $P_j \geq 0$, with M constraints of the form (3.10).

Using a similar argument, one can also simplify the coupled Riccati equations in Theorem 3.3.

Theorem 3.4 *Consider the jump linear system in (3.4) driven by a Markov chain $\theta[k]$ satisfying Assumption 3.1. Suppose that the Markov chain is i.i.d. so that $p_{ij} = q_j$ for all j. Also, suppose that the matrix $[B_i' \; D_{i2}']$ is injective for all i. Then,*

(a) The jump linear system (3.4) is MS detectable if and only if there exists a matrix $P \geq 0$, satisfying

$$P = \sum_{j=0}^{M-1} q_j \left[A_j P A_j' - E_{j1} G_j E_{j1}' + B_j B_j'\right] \tag{3.47}$$

where

$$\begin{aligned}
E_{j1} &= A_j P C_{j2}' + B_j D_{j2} \\
G_j &= (C_{j2} P C_{j2}' + D_{j2} D_{j2})^{-1}.
\end{aligned} \tag{3.48}$$

(b) Now, suppose (3.4) is MS detectable and $w[k]$ is zero-mean, white noise with unit variance independent of $\theta[k]$. Given a set of gain matrices

$$L = (L_0, \ldots, L_{M-1})$$

let $\hat{z}[k]$ be the estimator output in (3.19), and let $\sigma^2(L)$ be asymptotic mean-squared output estimation error (3.20). Then, there exists a $P \geq 0$, satisfying (3.47), such that the minimum estimation error is given by,

$$\min_L \sigma^2(L) = \sum_{i=0}^{M-1} q_i \mathbf{Tr}(C_{i1}PC'_{i1} + D_{i1}D'_{i1}),$$

where the minimization is over the set of MS stabilizing gain matrices L. Moreover, one set of minimizing gain matrices is given by

$$L_i = E_{i1}G_i$$

where E_{i1} and G_i are given in (3.48).

Proof: For part (a), first suppose that the system is MS detectable. From Theorem 3.3(a, there must exist matrices $P_j \geq 0$ satisfying (3.39). Since $p_{ij} = q_j$, $\overline{P}_j$ in (3.39) reduces to

$$\begin{aligned} \overline{P}_j &= \frac{1}{q_j}\sum_{i=0}^{M-1} p_{ij}q_iP_i = \frac{1}{q_j}\sum_{i=0}^{M-1} q_jq_iP_i \\ &= \sum_{i=0}^{M-1} q_iP_i. \end{aligned}$$

So, for all j, $\overline{P}_j = P$ where $P = \sum_i q_iP_i$. Using $\overline{P}_j = P$ along with (3.39),

$$P = \sum_{j=0}^{M-1} q_jP_j = \sum_{j=0}^{M-1} q_j \left[A_jPA'_j - E_{j1}G_jE'_{j1} + B_jB'_j\right].$$

So there exists a $P \geq 0$ satisfying (3.47).

Conversely, if there exists a $P \geq 0$ satisfying (3.47), we can take

$$P_j = q_j \left[A_jPA'_j - E_{j1}G_jE_{j1} + B_jB'_j\right].$$

Then a similar argument as in the proof of Proposition 3.6, shows that $\overline{P}_j \geq P$ for all j, and the matrices P_j satisfy (3.39). Therefore, we have shown that MS detectability is equivalent to the existence of a matrix $P \geq 0$ satisfying (3.47). This proves part (a).

Part (b) follows as a special case of Theorem 3.3(b) where $\overline{P}_i = P$ for all i. □

Theorem 3.4 shows that, in the i.i.d. case, the minimizing gain matrices, L_j, can be computed from a single Riccati equation (3.47) with a single unknown P. In contrast, the solution for general Markov chains involves M coupled Riccati equations (3.39) and M unknowns P_j.

Finally, the LMI estimation also simplifies for the i.i.d. case. For example, if the Markov chain is independent one can show that for the matrices W_j in Theorem 3.1, the components W_{j1} can be selected to be equal for all j. We will omit the proof, as it is almost identical to that of Proposition 3.6.

Theorem 3.5 *Consider the jump linear system in (3.4) driven by a Markov chain $\theta[k]$ satisfying Assumption 3.1 and $w[k]$ is zero-mean, white noise with unit variance independent of $\theta[k]$. Suppose the Markov chain is i.i.d. so that $p_{ij} = q_j$ for all i and j. Given sets of stabilizing gain matrices (L_1, L_2) as in (3.32), let $\hat{z}[k]$ be the estimator output in (3.31), and let $\sigma^2(L_1, L_2)$ be asymptotic mean-squared output estimation error (3.33). For any stabilizing sets of gain matrices (L_1, L_2), there exist $W_j \geq 0$ and $V_j \geq 0$ satisfying (3.34), (3.38) and the additional constraints*

$$W_{j1} = Q, \quad \textit{for all } j,$$

for some $Q \geq 0$, and

$$Q \geq \sum_{j=0}^{M-1} q_j \left([A_j' \ C_{j2}']\overline{W}_j[A_j' \ C_{j2}']' + [C_{j1}' C_{j2}']V_j[\ C_{j1}' \ C_{j2}']' \right), \tag{3.49}$$

where $\overline{W}_j$ is defined in (3.25).

Comparing this result to Theorem 3.5(b), we see that the assumption that Markov chain is independent significantly simplifies the estimation LMIs. Specifically, in the matrices W_j, the above theorem shows that all the components W_{j1} can be selected to be equal to a constant matrix Q. This greatly reduces the number of variables to search over. Also, the M LMI constraints in (3.35) are reduced to a single constraint (3.49).

3.9 Proofs

3.9.1 Proofs of Theorems 3.1 and 3.2

The proofs of Theorems 3.1 and 3.2 are almost identical to those of Theorems 2.3 and 2.4. We will only prove Theorem 3.1 here, since the modifications necessary for the proof of Theorem 3.2 are similar.

In the case of Theorem 3.1, the result is proven from Proposition 3.4, in a manner similar to the proof of Theorem 2.3 from Proposition 2.5. Specifically, for any estimator of the form (3.19), we define the error signals

$$e_x[k] = x[k] - \hat{x}[k], \quad e_z[k] = z[k] - \hat{z}[k].$$

Then, subtracting (3.1) and (3.19), we obtain the closed-loop system,

$$\begin{aligned} e_x[k+1] &= (A_i - L_i C_{i2}) e_x[k] + (B_i - L_i D_{i2}) w[k] \\ e_z[k] &= C_{i1} e_x[k] + D_{i1} w[k], \end{aligned} \tag{3.50}$$

when $\theta[k] = i$. Also, by the definition of $e_z[k]$,

$$\sigma^2(L) = \lim_{k \to \infty} \mathbf{E} \|z[k] - \hat{z}[k]\|^2 = \lim_{k \to \infty} \mathbf{E} \|e_z[k]\|^2.$$

Observe that the closed-loop system (3.50) is itself a jump linear system and, therefore, can be analyzed with Proposition 3.4.

To prove part (a), suppose there exist matrices W_i satisfying (3.24) and $\overline{W}_i \geq 0$. We will first show that $\overline{W}_{i1} > 0$. To this end, first note that since C_{i1} is injective, $C_{i1}' C_{i1} > 0$. Also, in (3.24), $\overline{W}_j \geq 0$ for all j, and therefore $W_{i1} \geq C_{i1}' C_{i1} > 0$ for all i. Equivalently, $W_{j1} > 0$ for all j. Since $p_{ij} \geq 0$ for all i and j, and

$$\sum_{j=0}^{M-1} p_{ij} = 1,$$

it follows that

$$\sum_{j=0}^{M-1} p_{ij} W_{j1} > 0.$$

The definition of $\overline{W}_i$ in (3.26) now shows that $\overline{W}_{i1} > 0$ for all i.

Since $\overline{W}_{i1} > 0$, we can define L_i as in (3.27). Also, let $Q_i = W_{i1}$. Then,

$$\overline{Q}_i = \overline{W}_{i1} = \sum_{j=0}^{M-1} p_{ij} Q_i.$$

Using the property (3.24), and the definition of L_i in (3.27), we can repeat the calculation in the proof of Theorem 2.3 to obtain

$$\begin{aligned}
&(A_i - L_i C_{i2})'\overline{Q}_i(A_i - L_i C_{i2}) + C_{i1}'C_{i1} \\
&= A_i'\overline{Q}_i A_i - C_{i2}'L_i'\overline{Q}_i A_i - A_i'\overline{Q}_i L_i C_{i2} + C_{i2}'L_i'\overline{Q}_i L_i C_{i2} + C_{i1}'C_{i1} \\
&= A_i'\overline{W}_{i1}A + C_{i2}'W_{i2}'A_i + A_i'W_{i2}C_{i2} + C_{i2}'W_{i2}'\overline{W}_{i1}^{-1}W_{i2}C_{i2} + C_{i1}'C_{i1} \\
&\le A_i'\overline{W}_{i1}A + C_{i2}'W_{i2}'A_i + A_i'W_{i2}C_{i2} + C_{i2}'W_{i3}C_{i2} + C_{i1}'C_{i1} \\
&= [A_i'\ C_{i2}']\overline{W}_i[A_i'\ C_{i2}']' + C_{i1}'C_{i1} \\
&\le W_{i1} = Q_i,
\end{aligned}$$

where we have used the Schur complement on the matrix $\overline{W}_i \ge 0$ for the property that

$$W_{i3} \ge W_{i2}'\overline{W}_{i1}^{-1}W_{i2}.$$

Hence, the matrices Q_i satisfy the Lyapunov LMIs for the closed-loop system. Also, $C_{i1}'C_{i1} > 0$ for all i. Therefore, by Proposition 3.4(b), the closed-loop system must be MS stable, and the matrices L_i are MS stabilizing. Moreover, using Proposition 3.4(b) and a calculation similar to that in the proof of Theorem 2.3, the asymptotic mean-squared error is bounded by

$$\begin{aligned}
\sigma^2(L) &\le \sum_{i=0}^{M-1} q_i \mathbf{Tr}\left((B_i - L_i D_{i2})'\overline{Q}_i(B_i - L_i D_{i2}) + D_{i1}'D_{i1}\right) \\
&= \sum_{i=0}^{M-1} q_i \mathbf{Tr}\left([B_i'\ D_{i2}']\overline{W}_i[B_i'\ D_{i2}']' + D_{i1}'D_{i1}\right),
\end{aligned}$$

which proves part (a).

Conversely, consider any MS stabilizing set of gain matrices,

$$L = (L_0, \ldots, L_{M-1}).$$

Then, the closed-loop system (3.50) must be MS stable. Consequently, by Proposition 3.4(a), there must exist matrices $Q_i \ge 0$, satisfying the Lyapunov equation for the closed-loop system,

$$Q_i = (A_i - L_i C_{i2})'\overline{Q}_i(A_i - L_i C_{i2}) + C_{i1}'C_{i1}, \tag{3.51}$$

with

$$\sigma^2(L) = \sum_{i=0}^{M-1} q_i \mathbf{Tr}\left((B_i - L_i D_{i2})'\overline{Q}_i(B_i - L_i D_{i2}) + D_{i1}'D_{i1}\right). \tag{3.52}$$

Similar to the proof of Theorem 2.3, we define,

$$W_{i1} = Q_i, \quad W_{i2} = -\overline{Q}_i L_i, \quad W_{i3} = L_i' \overline{Q}_i L_i, \tag{3.53}$$

we see that, $\overline{W}_{i1} = \overline{Q}_i$ so

$$\overline{W}_i = \begin{bmatrix} \overline{Q}_i & -\overline{Q}_i L_i \\ -L_i' \overline{Q}_i & L_i' \overline{Q}_i L_i \end{bmatrix} = [I \ \ -L_i]' \overline{Q}_i [I \ \ -L_i] \geq 0,$$

so $\overline{W}_i \geq 0$. Also, using (3.53) along with (3.51), it can be verified that W_i satisfies the LMI (3.24). Also combining (3.53) and (3.52) one can prove the bound in (3.29). This proves part (b).

3.9.2 Proof of Theorem 3.3

The proof is straightforward, but requires some technical notation. As usual, we will use the notation P and L to denote a set of variance and gain matrices,

$$\begin{aligned} P &= (P_0, \ldots, P_{M-1}), \\ L &= (L_0, \ldots, L_{M-1}). \end{aligned}$$

We will write $P \geq 0$ when $P_i \geq 0$ for all components $i = 0, \ldots, M-1$. Given a set of variance matrices $P \geq 0$ and gain matrices L, define the Lyapunov and Riccati operators,

$$\begin{aligned} \mathrm{Lyap}_i(P, L) &= P_i - \sum_{j=0}^{M-1} p_{ij} \left[A_{Lj} P_j A_{Lj}' + B_{Lj} B_{Lj}' \right], \\ \mathrm{Ric}_i(P) &= P_i - \sum_{j=0}^{M-1} p_{ij} \left[(A_j P_j A_j' - E_{1j}(P_j) G_j(P_j) E_{1j}(P_j)' + B_j B_j' \right], \end{aligned}$$

where

$$\begin{aligned} A_{Lj} &= A_j - L_j C_{j2} \\ B_{Lj} &= B_j - L_j D_{j2} \\ E_{1j}(P_j) &= A_j P_j C_{2j}' + B_j D_{2j}' \\ G_j(P_j) &= (C_{2j} P_j C_{2j}' + D_{2j} D_{2j}')^{-1} \end{aligned}$$

With these definitions, observe that the coupled Riccati equations (3.39) can be rewritten $\mathrm{Ric}_i(P) = 0$ for all i.

We now need two simple technical lemmas.

Lemma 3.1 *For any i and gain matrix L_i, $B_{Li}B'_{Li} > 0$.*

Proof: Observe that

$$B_{Li} = B_i - L_i D_{i2} = [I \ L_i][B'_i \ D'_{i2}]'.$$

Now, by the assumption of the theorem, $[B'_i \ D'_{i2}]$ is injective. Equivalently, $[B'_i \ D'_{i2}]'$ is onto. The matrix $[I \ L_i]$ is also onto for any L_i. Since the product of two onto matrices is onto, B_{Li} is onto, and therefore, $B_{Li}B'_{Li} > 0$.

Lemma 3.2 *For any set of matrices P and L with $P \geq 0$,*

$$Lyap_i(P, L) \leq Ric_i(P), \quad \forall i = 0, \ldots, M-1$$

with equality when $L_i = E_{i1}(P_i)G_i(P_i)$.

Proof: Fix any set of matrices P and L. To simplify the notation, let's write E_{j1} for $E_{j1}(P_j)$ and G_j for $G_j(P_j)$. Now, using the definitions for A_{Lj}, B_{Lj}, E_{j1} and G_j, it can be verified that for all j,

$$\begin{aligned}
&A_{Lj}P_jA'_{Lj} + B_{Lj}B'_{Lj} \\
&\quad = \quad A_jP_jA'_j + B_jB'_j - E_{j1}L'_j - L_jE'_{j1} + L_jG_j^{-1}L'_j \\
&\quad = \quad A_jP_jA'_j + B_jB'_j - E_{j1}G_jE'_{j1} \\
&\quad + \quad (L_j - E'_{j1}G_j)G_j^{-1}(L_j - E'_{j1}G_j)' \\
&\quad \geq \quad A_jP_jA'_j + B_jB'_j - E_{j1}G_jE'_{j1},
\end{aligned}$$

with equality in the last step when $L_j = E_{j1}G_j$. It follows that for all i, $\text{Lyap}_i(P, L) \leq \text{Ric}_i(P)$ with equality occurring when $L_j = E_{j1}G_j$.

Proof of Theorem 3.3: For part (a), first suppose that there exist matrices $P \geq 0$ satisfying the coupled Riccati equations (3.39). Equivalently, $\text{Ric}_i(P) = 0$ for all i. By Lemma 3.2, if we let $L_i = E_{i1}(P_i)G_i(P_i)$,

$$\text{Lyap}_i(P, L) = \text{Ric}_i(P) = 0.$$

Also, by Lemma 3.1, $B_{Li}B'_{Li} > 0$ for all i. Therefore, from Proposition 3.3(b), the matrices $A_{Li} = A_i - L_iC_{i2}$ must be MS stable. Therefore, (A_i, C_{i2}) is MS detectable.

Conversely, suppose (A_i, C_{i2}) is MS detectable. We want to show that there exist matrices $P \geq 0$ with $\text{Ric}_i(P) = 0$ for all i. Define the set,

$$X = \{(P, L) \ : \ \text{Lyap}_i(P, L) \geq 0 \text{ for all } i\}.$$

We first claim that X is non-empty. Since (A_i, C_{i2}) is MS detectable, there exist matrices L_i such that $A_i - L_i C_{i2}$ is MS stable. By Proposition 3.3(a), there must exist matrices $P_i \geq 0$ with $\text{Lyap}_i(P, L) \geq 0$. Therefore, X is non-empty.

Now, let

$$J(P) = \sum_{i=0}^{M-1} q_i \mathbf{Tr}(C_{i1} P_i C_{i1}' + D_{i1} D_{i1}'),$$

and let $(P^*, L^*) \in X$ be any matrices satisfying the following two conditions: For any $(P, L) \in X$,

(a) $J(P) \geq J(P^*)$,

(b) If $J(P) = J(P^*)$, then $\mathbf{Tr}(P) \geq \mathbf{Tr}(P^*)$, where $\mathbf{Tr}(P)$ is defined as

$$\mathbf{Tr}(P) = \sum_{i=0}^{M-1} \mathbf{Tr}(P_i).$$

To find such a (P^*, L^*) we first consider the set of minimizers of $J(P)$ subject to $(P, L) \in X$. Any element (P^*, L^*) in this minimizing set will satisfy condition (a). To satisfy condition (b), we select (P^*, L^*) amongst the minimizers of $J(P)$ to be the element with the minimum total trace, $\mathbf{Tr}(P^*)$.

By Lemma 3.2,

$$\text{Ric}_i(P^*) = \text{Lyap}_i(P_i^*, E_{i1}(P^*) G_i(P^*)) \leq \text{Lyap}_i(P^*, L^*).$$

So, without loss of generality we can assume that minimizing element $(P^*, L^*) \in X$ satisfies $L^* = E_{i1}(P^*) G_i(P^*)$. Under this assumption, we claim that $\text{Ric}_i(P^*) = 0$, so that P^* is the desired solution to the coupled Riccati equations.

To prove that P^* satisfies the coupled Riccati equations, let $Q_i = \text{Ric}_i(P^*) = \text{Lyap}_i(P^*, L^*)$. We want to show $Q_i = 0$. Since $(P^*, L^*) \in X$, $Q_i \geq 0$. Now let

$$\begin{aligned} A_{Lj}^* &= A_j - L_j^* C_{j2} \\ B_{Lj}^* &= B_j - L_j^* D_{j2}. \end{aligned}$$

Then, the equation $\text{Lyap}_i(P^*, L^*) = Q_i$ expands to

$$P_i^* = \sum_{j=0}^{M-1} p_{ij} \left[A_{Lj}^* P_j^* A_{Lj}^{*'} + B_{Lj}^* B_{Lj}^{*'} + Q_j \right]. \tag{3.54}$$

Now Lemma 3.1 shows that $B^*_{Li}B^{*'}_{Li} > 0$ for all i. Therefore, by Proposition 3.3(b), $A^*_{Lj} = A_i - L^*_i C_{i2}$ must be MS stable. So, by Proposition 3.1(a), there exist matrices $S_i \geq Q_i$ with

$$S_i = \sum_{j=0}^{M-1} p_{ij} \left[A^*_{Lj} S_j A^{*'}_{Lj} + Q_j \right], \tag{3.55}$$

with $S_i \geq Q_i$. Subtracting (3.55) from (3.54), we obtain,

$$P^*_i - S_i = \sum_{j=0}^{M-1} p_{ij} \left[A^*_{Lj}(P^*_j - S_j)A^{*'}_{Lj} + B^*_{Lj}B^{*'}_{Lj} \right],$$

so $\text{Lyap}_i(P^* - S, L^*) = 0$ for all i. Therefore, $(P^* - S, L^*) \in X$. By condition (a) of the definition of (P^*, L^*), $J(P^*) \leq J(P^* - S)$. But, since $S_i \geq 0$,

$$J(P^*) - J(P^* - S) = \sum_{i=0}^{M-1} q_i \mathbf{Tr}(C_{i1} S_i C'_{i1}) \geq 0$$

so $J(P^* - S) \leq J(P^*)$. Therefore, $J(P^*) = J(P^* - S)$. It follows from condition (b) of the definition of (P^*, L^*) that $\mathbf{Tr}(P^*) = \mathbf{Tr}(P^* - S)$, or equivalently, that $\mathbf{Tr}(S) = 0$. But, since $S_i \geq Q_i \geq 0$,

$$0 \leq \mathbf{Tr}(Q_i) \leq Tr(S_i) \leq \mathbf{Tr}(S) = 0,$$

so $\mathbf{Tr}(Q_i) = 0$. Since $Q_i \geq 0$, this implies that $Q_i = 0$, which completes the proof of part (a).

For part (b), consider the estimator (3.19) defined for some stabilizing gain matrices L. As in the proof of Theorem 3.1, we define the error signals

$$e_x[k] = x[k] - \hat{x}[k], \quad e_z[k] = z[k] - \hat{z}[k],$$

and obtain the closed-loop system that, when $\theta[k] = i$,

$$\begin{aligned} e_x[k+1] &= A_{Li} e_x[k] + B_{Li} w[k] \\ e_z[k] &= C_{i1} e_x[k] + D_{i1} w[k], \end{aligned}$$

where, as usual, $A_{Li} = A_i - L_i C_{i2}$ and $B_{Li} = B_i - L_i D_{i2}$. Also, by the definition of $e_z[k]$,

$$\sigma^2(L) = \lim_{k \to \infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2 = \lim_{k \to \infty} \mathbf{E}\|e_z[k]\|^2.$$

Now, if L is a set of stabilizing gain matrices, A_{Li} is MS stabilizing, and therefore Proposition 3.3 shows that

$$\sigma^2(L) = \sum_{i=0} q_i \mathbf{Tr}(C_{i1} P_i C'_{i1} + D_{i1} D'_{i1}),$$

where $P_i \geq 0$ is the solution to the coupled Lyapunov equations

$$P_i = \sum_{j=0}^{M-1} p_{ij} \left[A_{Lj} P_j A'_{Lj} + B_{Lj} B'_{Lj} \right].$$

Using the notation in this proof, we can say that

$$\sigma^2(L) = J(P)$$

where P is the solution to the Lyapunov equations, $\text{Lyap}_i(P, L) = 0$. Thus,

$$\min_L \sigma^2(L) = \min_{L,P} J(P)$$

where the second minimum is over (L, P) satisfying $\text{Lyap}_i(P, L) = 0$. But, by the construction of (P^*, L^*), (P^*, L^*) is precisely the minimizing solution. Therefore,

$$\min_L \sigma^2(L) = \sum_{i=0} q_i \mathbf{Tr}(C_{i1} P_i^* C'_{i1} + D_{i1} D'_{i1}),$$

with the minimum occurring with $L_i^* = E_{1i}(P_i^*) G_i(P_i^*)$.

Chapter 4

Estimation with Markovian Losses

4.1 Introduction

The results in the previous chapter are somewhat general and abstract. To illustrate the LMI analysis more concretely, this chapter considers a simple application to signal reconstruction from lossy data. For illustration, we will focus on signal reconstruction with additive noise and erasure losses. Of course, the jump linear model is extremely general, and significantly more general loss and signal models can also be considered. Some of these extensions will be briefly discussed in Section 4.6.

The basic estimation problem we consider is shown in Figure 4.1. The signal to be estimated is denoted $z[k]$, and is modeled as a correlated random process. The signal is corrupted by additive noise $d[k]$, to yield a noisy signal $y_0[k]$. The noisy signal, $y_0[k]$, is then transmitted over a channel that erases some of the samples. The channel output is

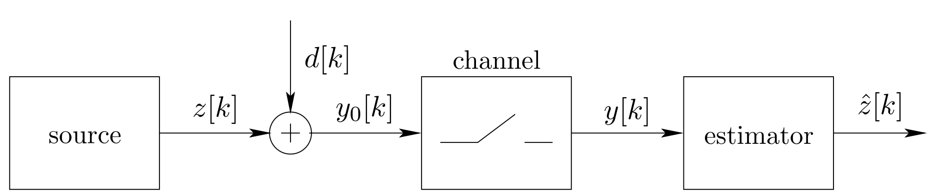

Figure 4.1. Lossy data estimation problem.

denoted $y[k]$, and is given by

$$y[k] = \begin{cases} y_0[k] & \text{if sample } k \text{ is not lost} \\ 0 & \text{if sample } k \text{ is lost.} \end{cases} \tag{4.1}$$

The problem is to estimate the original signal $z[k]$ from the samples of the noisy signal $y_0[k]$ that are not lost in the channel. The problem is identical to the denoising problem described in Section 2.5, except that some of the samples of $y_0[k]$ may be erased and are therefore unavailable to the estimator.

The estimation of $z[k]$ from $y[k]$ is easily posed as a Kalman filtering problem: If we model $z[k]$ as a stationary random process, then one can write both the unknown signal, $z[k]$, and observed channel outputs, $y[k]$ as outputs of a single linear system driven with noise. The system will have time variations arising from the sample erasures. With this model, the Kalman filter provides the optimal estimate of $z[k]$ given the channel output $y[k]$. Assuming a Markov model for the erasure process, the LMI analysis in the previous chapter can then be used to quantify the asymptotic average estimator performance of the Kalman estimator. In this way, the LMI analysis provides a method for computing the estimator performance as a function of the statistics of the signal to be estimated, $z[k]$, the additive noise, $d[k]$, and statistics of the channel erasure process.

For communication problems, which is our main interest, the analysis here can be used to compute the average reconstruction error in estimating a signal transmitted over a lossy channel. The Markov process could model the loss process in the channel and the additive noise could the quantization noise. This sort of analysis will be used throughout Chapters 6 and 7, which consider the transmission of quantized data over erasure channels.

4.1.1 Related Work in Sensor Networks

It is well-known that the Kalman filter can be used for estimation with intermittent observations. Such applications have appeared at least as early as the work of Nahi [114]. However, there are relatively few methods currently available for quantifying the asymptotic expected error variance of the Kalman estimator as a function of the loss statistics.

The most recent work concerned with this problem has arisen in the study of lossy sensors [141]. In this application, $z[k]$ represents the input to a sensor and $y[k]$ represents the sensor output. The signal $d[k]$ represents sensor noise and accounts for any imperfections in the sensor. The losses of the sensor data can model

- Losses due to communication failures that arise when the sensor is remotely located and must communicate to the sensor processing unit,
- Intermittent failures of the sensor itself,
- Cases where the sensor may transmit data, by design, at some irregular intervals.

The study of such lossy sensors has recently received significant attention in the context of so-called sensor networks. It has been theorized that sensor networks will be composed of a large number of inexpensive sensors to monitor a large number of physical quantities, possibly spread over a large geographical area. Due to the numbers of sensors required, the individual sensors must be made inexpensive and are constrained in their reliability, power and communication abilities. Consequently, data from such sensors may be lossy and it is useful to quantify the performance degradation due to losses.

In [139, 140], Sinopoli *et al.* considered the special case of i.i.d. erasures with an erasure probability λ. The work shows that the asymptotic state error variance can be bounded by solving a modified form of the standard algebraic Riccati equation, which can, in turn, be solved through an LMI optimization. Using this LMI analysis, Sinopoli shows that for unstable systems, there is a critical error probability, λ_{crit}, above which the expected error is infinite.

Sinopoli's work has recently been extended by Liu and Goldsmith [103], who consider the case of two linear observations of the state, each being erased with independent probabilities λ_1 and λ_2. Using an LMI analysis similar to Sinopoli's, they are able to derive a region of (λ_1, λ_2) where the asymptotic error variance is finite, and for (λ_1, λ_2) in that region, they provide a bound on the expected error.

In this chapter, we will consider a general linear-Markov model from which the case of single and multiple observations with i.i.d. erasures follow as special cases. In the special cases of Sinopoli [140]and Liu and Goldsmith [103], we are able to derive similar LMIs to bound the asymptotic expected error. The modified algebraic Riccati equation of Sinopoli is also shown to be a special case of the coupled algebraic Riccati equations in Section 3.7.

However, the framework presented here also extends the lossy sensor analysis of Sinopoli and Liu and Goldsmith. For one thing, the LMI analysis can be applied to general Markov erasures, thus allowing us to consider correlated and more complicated erasure processes. The framework can also consider arbitrary linear time-varying channels instead of simple erasure processes, and can painlessly handle arbitrary multiple linear observations. In

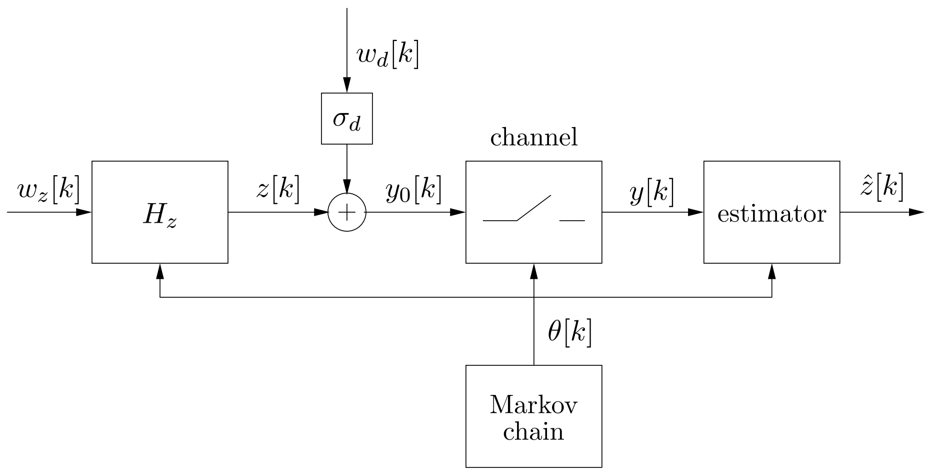

Figure 4.2. Jump linear model for lossy data estimation problem.

summary, the framework can serve as a global platform on which to address a plethora of lossy sensor problems with an array of realistic, rich signal and loss models.

4.2 Jump Linear Modeling

To employ the LMI framework of the previous chapter to the lossy sensor problem, we need to describe the observed signal, $y[k]$, and the signal to estimated, $z[k]$, as a jump linear system. One such model is shown in Figure 4.2. The modeling is similar to the state space modeling used for the denoising problem in Section 2.5. As before, the signal to be estimated is modeled as white noise filtered by an LTI system H_z. We will assume that H_z admits a state space representation,

$$\begin{aligned} x[k+1] &= Ax[k] + B_z w_z[k] \\ z[k] &= Cx[k] + D_z w_z[k], \end{aligned} \tag{4.2}$$

where $w_z[k]$ is zero-mean, white noise with unit variance, and the system matrices (A, B_z, C, D_z) are chosen to match the statistics of $z[k]$. We will assume that the noise $d[k]$ can be described by,

$$d[k] = \sigma_d w_d[k], \tag{4.3}$$

where $w_d[k]$ is also zero-mean, unit-variance white noise and σ_d is a scaling term to account for the noise variance. Since the noise $d[k]$ is additive, the channel input is given by,

$$y_0[k] = z[k] + d[k] = z[k] + \sigma_d w_d[k]. \tag{4.4}$$

The modeling of the signals $z[k]$ and $y_0[k]$ is identical to the modeling for the LTI denoising problem considered earlier in Section 2.5. The jump linear aspect of the problem considered here, comes in modeling the channel erasures. We will assume that channel erasures can be described by a Markov chain. Specifically, we will assume that there is an underlying Markov chain $\theta[k] \in \{0, \ldots, M-1\}$, and the erasures occur when the Markov chain enters a subset of states, $I_{loss} \subseteq \{0, \ldots, M-1\}$. The erasure channel output is given by,

$$y[k] = \begin{cases} y_0[k] & \text{if } \theta[k] \notin I_{loss} \\ 0 & \text{if } \theta[k] \in I_{loss}. \end{cases} \tag{4.5}$$

The Markov dynamics can also be easily incorporated in the source model to capture discrete changes in the signal statistics or observation noise variance.

This Markov model is extremely general and captures a large range of erasure processes:

- Independent erasures: In this case, there are $M = 2$ states, so that $\theta[k] = 0$ or 1. The sample $y_0[k]$ is lost when $\theta[k]$ enters one of the states, say $\theta[k] = 1$. That is, $I_{loss} = \{1\}$. To model independent erasures, we take the transition probabilities,

  $$\begin{aligned} p_{01} &= p_{11} = \lambda, \\ p_{00} &= p_{10} = 1 - \lambda. \end{aligned}$$

  With these transition probabilities, the Markov chain, and consequently, the erasure probability is i.i.d. with an erasure probability of λ. The steady state distribution is given by,

  $$(q_0, q_1) = (1 - \lambda, \lambda).$$

- Gilbert-Elliot erasures: This is a slight generalization of the i.i.d. erasures. As in the i.i.d. case, there are two states so that $\theta[k] = 0$ or 1, with the sample $y_0[k]$ being erased when $\theta[k] = 1$. In the Gilbert-Elliot model, the transition probabilities are given by

  $$\begin{aligned} p_{01} &= 1 - p_{00} = \lambda_1, \\ p_{10} &= 1 - p_{11} = \lambda_2, \end{aligned}$$

  for some parameters λ_1 and λ_2. The probability λ_1 represents the probability of a transition from the "good," or non-erased state, $\theta[k] = 0$, to the "bad", or erasure state, $\theta[k+1] = 1$. Similarly, λ_2 is the transition probability from the erasure state back to the good state. When $\lambda_1 = 1 - \lambda_2 = \lambda$, the Gilbert-Elliot models reduces to the i.i.d. erasure model with an erasure probability λ. However, using other values of λ_1 and λ_2, provides a simple model of correlated erasures. For example, if one takes λ_1 and λ_2 small, while in the good state, the erasures will occur infrequently, but when the erasures do occurs, it is likely that several consecutive samples will be erased. In general, for any λ_1 and λ_2, the stationary distribution is given by,

$$(q_0, q_1) = \left(\frac{\lambda_2}{\lambda_1 + \lambda_2}, \frac{\lambda_1}{\lambda_1 + \lambda_2} \right).$$

- Fixed Length Burst Losses: In this model, the samples are erased in consecutive groups of L samples at a time, with the probability of beginning such a loss being λ. We can model this loss process using an $M = L+1$ state Markov chain as follows: As before, we will let the state $\theta[k] = 0$ represent a non-loss state. For the other states $i = 1, \ldots, L$, $\theta[k] = i$ represents the state where the channel is in the i-th sample of the L-length loss sequence. To model the fact that the loss sequence can restart at any time with probability λ, we set the transition probability $p_{i1} = \lambda$ for all i. If the Markov chain is in state 0, and a new loss does not occur, it will remain in state 0. Otherwise, if a new loss sequence does not occur and the Markov chain is in state 1, it will transition to state 2. From state 2 it will move to state 3, and so on, up to state L from which it will return to state the non-loss state 0. The transition probabilities are given by

$$p_{ij} = \begin{cases} \lambda & \text{when } j = 1, \ i = 0, \ldots, L \\ 1 - \lambda & \text{when } (i, j) = (0, 0) \text{ or } (0, L) \\ 1 - \lambda & \text{when } j = i + 1, i = 1, \ldots, L - 1 \\ 0 & \text{otherwise} \end{cases}$$

4.3 LMI Analysis

Using the above linear model for the signals $z[k]$ and $y[k]$, and the Markov model for the erasure process, the estimation problem can be easily placed into the framework of jump linear estimation discussed in Chapter 3. To this end, we first combine (4.2) and (4.4), to rewrite the unknown signal, $z[k]$, and channel input, $y_0[k]$, as the outputs of a single linear state space system,

$$\begin{aligned} x[k+1] &= Ax[k] + Bw[k] \\ z[k] &= C_1 x[k] + D_1 w[k] \\ y_0[k] &= C_2 x[k] + D_2 w[k], \end{aligned} \tag{4.6}$$

where $w[k] = [w_z[k] \; w_d[k]]$ is a noise vector containing both the source noise $w_z[k]$ and additive noise $w_d[k]$, and the system matrices in (4.6) are given by

$$\begin{aligned} B &= [B_z \; 0], \\ C_1 &= C, \quad D_1 = [D_z \; 0], \\ C_2 &= C, \quad D_2 = [D_z \; \sigma_d]. \end{aligned}$$

Then, using (4.1), we can write that when $\theta[k] = i$, the channel output $y[k]$ is given by

$$y[k] = C_{i2}x[k] + D_{i2}w[k] \tag{4.7}$$

where

$$(C_{i2}, D_{i2}) = \begin{cases} (C_2, D_2) & \text{when } \theta[k] \notin I_{loss} \\ (0, 0) & \text{when } \theta[k] \in I_{loss} \end{cases}$$

Combining (4.7) and (4.6), we obtain the jump linear system: when $\theta[k] = i$,

$$\begin{aligned} x[k+1] &= Ax[k] + Bw[k] \\ z[k] &= C_1x[k] + D_1w[k] \\ y[k] &= C_{i2}x[k] + D_{i2}w[k]. \end{aligned} \tag{4.8}$$

In this system, the unknown signal to be estimated, $z[k]$, and the observed channel output, $y[k]$, are described as the outputs of a single jump linear system. Consequently, the estimation of $z[k]$ from the channel output, $y[k]$, is precisely the jump linear estimation problem considered in Chapter 3.

Following the analysis there, we can consider a jump linear estimator of the form: when $\theta[k] = i$,

$$\begin{aligned} \hat{x}[k+1] &= A\hat{x}[k] + L_i(y[k] - C_{i2}\hat{x}[k]) \\ \hat{z}[k] &= C_1\hat{x}[k] \end{aligned} \tag{4.9}$$

defined for some set of gain matrices

$$L = (L_0, \ldots, L_{M-1}).$$

The estimator (4.9) is itself a jump linear system and produces a state estimate $\hat{x}[k]$ and output estimate $\hat{z}[k]$. In (4.9), we have written the equations for the strictly causal estimator. A causal estimator can also be considered as discussed in Section 3.6.2.

The set of gain matrices L_i in (4.9) are considered design parameters and can be selected to minimize the asymptotic mean-squared error,

$$\sigma^2(L) = \lim_{k \to \infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2,$$

where the dependence on L is through the estimate $\hat{z}[k]$ in (4.9). The minimization

$$\min_L \sigma^2(L)$$

can be performed via the LMI in Section 3.6 to find the optimal gain matrices and the corresponding minimum estimation error.

Using the definition of the system matrices B, C_1, C_2, etc., we can simplify the estimator equations (4.9). For example, when $\theta[k] \in I_{loss}$, the estimator updates takes the form:

$$\begin{aligned} \hat{x}[k+1] &= A\hat{x}[k] \\ \hat{z}[k] &= C\hat{x}[k] \end{aligned} \tag{4.10}$$

Here, we have used the fact that when there is a sample erasure, $y[k] = 0$ and $C_{i2} = 0$. Therefore the correction term $L_i(y[k] - C_{i2}\hat{x}[k]) = 0$. Consequently, the gain matrices L_i play no role for the states $i \in I_{loss}$. When there is no loss, so that $\theta[k] = i \notin I_{loss}$, we have the update

$$\begin{aligned} \hat{x}[k+1] &= A\hat{x}[k] + L_i(y[k] - C\hat{x}[k]) \\ \hat{z}[k] &= C\hat{x}[k] \end{aligned} \tag{4.11}$$

4.4 Independent Losses

While the LMI framework can be applied to the lossy sensor problem with general Markov erasures, it is useful to briefly consider the special case of i.i.d. erasures. As discussed in Section 4.2, i.i.d. erasures can be modeled with $M = 2$ states, so that $\theta[k] = 0$ or 1. We will let state 0 correspond to the non-loss state, and state 1 to the erasure state. For i.i.d. erasures, the transition probabilities are given by,

$$\begin{aligned} p_{01} &= p_{11} = \lambda, \\ p_{00} &= p_{10} = 1 - \lambda, \end{aligned} \tag{4.12}$$

where λ is the probability of transitioning to the erasure state $\theta[k] = 1$.

Now, we saw that in the jump linear estimator (4.10) and (4.11), that the gain matrices L_i do not play any role for the states $i \in I_{loss}$. Therefore, since the i.i.d. erasure model has only one non-erasure state, the estimator is described by one gain matrix, which we will denote by L. The estimator reduces to:

$$\begin{aligned} \hat{x}[k+1] &= A\hat{x}[k] \\ \hat{z}[k] &= C\hat{x}[k] \end{aligned} \tag{4.13}$$

when $\theta[k]$ is in the erasure state $\theta[k] = 1$, and

$$\begin{aligned} \hat{x}[k+1] &= A\hat{x}[k] + L(y[k] - C\hat{x}[k]) \\ \hat{z}[k] &= C\hat{x}[k] \end{aligned} \tag{4.14}$$

when $\theta[k]$ is in the non-loss state $\theta[k] = 0$. As before, we can regard the asymptotic mean-squared error

$$\sigma^2(L) = \lim_{k\to\infty} \mathbf{E}\|z[k] - \hat{z}[k]\|^2, \tag{4.15}$$

as a function of the gain matrix L. The gain matrix L minimizing the error can be found as a special case of the LMI optimization in Section 3.6.

In addition to performing the LMI analysis, following Section 3.7, one can also derive the coupled Riccati equations for the lossy sensor systems. In the special case of i.i.d. erasures, these equations take on a particularly simple form as shown in the following Theorem.

Theorem 4.1 *Consider the lossy sensor system in Figure 4.1, where $z[k]$ and $y[k]$ are described by the state space system (4.6), and the channel output is described by (4.1). Suppose that the Markov chain $\theta[k]$ is an i.i.d. erasure process described above for some loss probability λ. For a gain matrix L, let $\hat{z}[k]$ be the estimate given in (4.13) and (4.14), and let $\sigma^2(L)$ be the corresponding asymptotic mean-squared error in (4.15). Assume $[B'\ D_2']$ is injective. Then,*

(a) The (4.6) is MS detectable (i.e. there exists an MS stabilizing gain matrix L) if and only if there exists a matrix $P \geq 0$, satisfying

$$P = APA' - (1-\lambda)E_1GE_1' + BB' \tag{4.16}$$

where

$$\begin{aligned} E_1 &= APC' + BD_2' \\ G &= (CPC' + D_2D_2)^{-1}. \end{aligned} \tag{4.17}$$

(b) Suppose that (4.6) is MS detectable and $w[k]$ is zero-mean, white noise with unit variance independent of $\theta[k]$. Then, there exists a $P \geq 0$, satisfying (4.16), such that the minimum estimation error is given by,

$$\min_L \sigma^2(L) = \mathbf{Tr}(CPC' + D_1D_1'),$$

where the minimization is over the set of MS stabilizing gain matrices L. Moreover, one minimizing gain matrix is given by

$$L_i = E_1 G$$

where E_1 and G are given in (4.17).

Proof: The theorem is simply a restatement of Theorem 3.4 for the special case of the lossy sensor problem. Specifically, Theorem 3.4(a) shows that the system (4.6) is MS detectable if and only there exists a $P \geq 0$ satisfying the Riccati equation (3.47). Now, the sensor loss system has two states: $i = 0$ or 1. For the non-loss state, $i = 0$:

$$E_{i1} G_i E_{i1}' = E_1 G E_1',$$

where E_1 and G are defined in (4.17). For the loss state, $i = 1$, $C_{i2} = 0$ and $D_{i2} = 0$, so

$$E_{i1} G_i E_{i1}' = 0.$$

Substituting these expressions, along with the definition in A_i, B_i, etc. and the probability distribution $(q_0, q_1) = (1 - \lambda, \lambda)$ into (3.47), we obtain,

$$\begin{aligned} P &= q_0 \left[APA' + BB' - E_{01} G_0 E_{01}' \right] \\ &+ q_1 \left[APA' + BB' - E_{11} G_1 E_{11}' \right] \\ &= (1 - \lambda) \left[APA' + BB' - E_1 G E_1' \right] + \lambda \left[APA' + BB' \right] \\ &= APA' - (1 - \lambda) E_1 G E_1' + BB', \end{aligned}$$

which is precisely (4.16). We can thus conclude that the system (4.6) is MS detectable if and only if there exists a $P \geq 0$ satisfying (4.16).

Now suppose that the system is MS detectable. In this case, Theorem 3.4(b) shows that there exists a $P \geq 0$ satisfying (4.16) with

$$\begin{aligned} \min_L \sigma^2(L) &= q_0 \mathbf{Tr}(CPC' + D_1 D_1') + q_1 \mathbf{Tr}(CPC' + D_1 D_1') \\ &= \mathbf{Tr}(CPC' + D_1 D_1'). \end{aligned}$$

Finally, the minimizing gain matrix L for the non-loss state is, $L = -E_1 G$. $\square$

4.5 Algebraic Riccati Equation for I.I.D. Losses

The analysis in the previous section results in a modified form of the standard algebraic Riccati equation,

$$\begin{aligned} P &= APA' - (1-\lambda)E_1 G E_1' + BB', \\ E_1 &= APC' + BD' \\ G &= (CPC' + DD')^{-1} \end{aligned} \tag{4.18}$$

for a loss parameter λ. When $\lambda = 0$, the equation reduces to the standard algebraic Riccati equation (2.17), and when $\lambda = 1$, it reduces to the discrete Lyapunov equation (2.6). A similar equation (with $BD' = 0$) is derived in [140], which also looks at the lossy sensor problem with i.i.d. erasures. Following the terminology there, we will call (4.18) the *modified algebraic Riccati equation*.

As discussed in Section 3.7, it is difficult to solve the coupled algebraic Riccati equations related to the estimation problem of a general jump linear system. Typically, one must solve the LMI optimization in Section 3.6 to find the optimal estimation gain matrices. However, in the special case of the lossy sensor estimation problem with i.i.d. erasures, the coupled Riccati equations reduce to (4.18), which can also be solved with the following simple iterative procedure.

Suppose D has a scalar output. That is, D is a row vector. For $\sigma > 0$, define the matrix function,

$$D(\sigma) = [D\ \sigma].$$

Then, it is easy to verify that P is a solution to (4.18) if and only if there exists a $\sigma > 0$ satisfying the coupled equations,

$$P = APA' - E_1(CPC' + D(\sigma)D(\sigma)')^{-1}E_1' + BB' \tag{4.19}$$

$$E_1 = APC' + BD' \tag{4.20}$$

$$\sigma^2 = \frac{\lambda}{1-\lambda}(CPC' + DD'). \tag{4.21}$$

For a fixed σ, the solution to equations (4.19) and (4.20) is a standard algebraic Riccati equation and can be easily solved for P. Also, it can be verified that the solution P monotonically increases with σ. This leads to the following iterative bisection search procedure.

- *Step 1:* Find minimum and maximum values, σ_{min} and σ_{max}, for the possible solution σ.

- *Step 2:* Set σ to the midpoint of the interval, $\sigma = 1/2(\sigma_{min} + \sigma_{max})$.

- *Step 3:* Solve the algebraic Riccati equation (4.19) and (4.20) for P using the midpoint value of σ.

- *Step 4:* If

$$\sigma^2 > \frac{\lambda}{1-\lambda}(CPC' + DD')$$

  then set $\sigma_{max} = \sigma$ and return to Step 2. Otherwise, set $\sigma_{min} = \sigma$ and return to Step 2

This iterative procedure will converge exponentially to a σ and P satisfying the coupled equations (4.19) and (4.21).

4.6 Extensions

Up to now, we have presented the lossy sensor problem with a single observation undergoing Markov erasures. However, the general LMI estimation theory in Chapter 3 is extremely powerful, and can be used to extend the lossy sensor analysis in a number of ways:

- *Soft erasures:* The above model considers only erasures, where the sample $y[k]$ is completely lost. However, we may be interested in so-called "soft" erasures where the observed data is intermittently degraded but not completely erased. One simple model for such losses is to have varying levels of additive noise $d[k]$, where the degraded samples are subject to higher noise levels than the undegraded samples. This kind of degradation is easily modeled in the jump linear framework by having different values for the output matrix D_{i2} depending on the sate i.

- *Sensor fusion and multiple observations:* The jump linear framework can also easily incorporate multiple observations. Multiple observations may arise in sensor networks, where a single quantity is observed with multiple sensors. Each sensor may have different physical dynamics and, possibly different communication channels to the estimator. The combining of data from multiple observations is often called *sensor fusion*.

To incorporate multiple observations in the jump linear model, we simply use a vector-valued observation signal, $y[k]$, where each component, $y_j[k]$, represents the observation from a single sensor. Using a Cartesian product of state spaces, one can create a single Markov chain to model the dynamics of multiple channels. The Markov chain can also model random selections of sensor data. Using this model, the jump linear framework can quantify the benefit of multiple observations and study the effect of losses in sensor networks.

- *Estimation of multiple signals:* The jump linear model can similarly be used to study the estimation of multiple correlated random processes. For this purpose, we use a vector-valued signal $z[k]$ whose components are the various signals to be estimated. The correlations of the various components of $z[k]$ can be easily captured with a general linear model that can describe an arbitrary autocorrelation matrix.

- *Stability analysis:* The analysis in both Sinopoli [139] and Liu and Goldsmith [103] consider potentially unstable systems, and determine under what loss probabilities is the expected error finite. Their analysis provides upper and lower bounds on the maximum error probability, which they show in simulation to be close to one another in practice. While the analysis here cannot provide a upper bound on the maximum error probability, a similar lower bound can be derived. Specifically, for any loss probability, if the LMI is able to produce a set of stabilizing gain matrices, then the estimation error will be finite.

4.7 Numerical Examples

4.7.1 Denoising with Lossy Observations

As a simple numerical example, we consider the estimation of a lowpass signal from lossy data with various erasure models. For the model of the signal $z[k]$, we select the matrices (A, B_z, C, D_z) in (4.2) to correspond to a lowpass filter with cutoff frequency at 0.2 times the sample rate. Any filter can be used to design such a signal, and here we use the Chebyshev filter design routine provided in Matlab. The filter output is normalized so that $\mathbf{E}\|z[k]\|^2 = 1$. The additive noise $d[k]$ in (4.4) is set to 10 dB below the signal level.

For the erasure process, we consider two models: the i.i.d. erasures and fixed-length burst losses with a length of 3. The transition probabilities of both processes are discussed

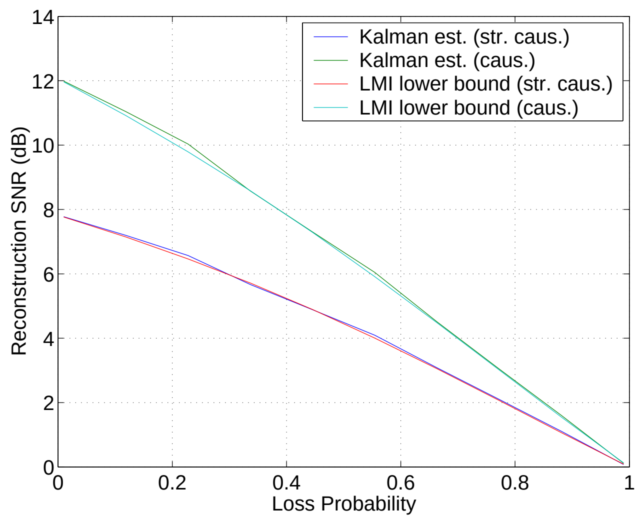

Figure 4.3. Reconstruction SNR of a lowpass signal from noisy samples with i.i.d. losses. Plotted are the strictly causal and causal estimates for the optimal Kalman filter, and the filter designed with the LMI method.

in Section 4.2. The i.i.d. model has two Markov states, and the fixed-length burst model has four states.

The results of the estimation of the signal $z[k]$ from the lossy data $y[k]$ for the two models are shown in Figure 4.3 and 4.4. For each model, we vary the loss probability and measure the performance of both the optimal time-varying Kalman filter, and the suboptimal jump linear estimator where the gain matrices are optimized with the LMI analysis. Both the strictly causal and causal estimator and simulated for each estimator. In each case, we measure the reconstruction SNR which is defined as

$$\text{SNR} = 10 \log_{10} \left[\frac{\mathbf{E}\|z[k]\|^2}{\mathbf{E}\|z[k] - \hat{z}[k]\|^2} \right].$$

It can be seen that for both erasure models, the LMI-based estimator matches the performance of the optimal Kalman estimator extremely well. For most loss probabilities, the gap is within a fraction of a dB.

4.7.2 Two-Dimensional Tracking

For a slightly more complex example, we consider the tracking of a two-dimensional signal

$$z[k] = [z_1[k]\ z_2[k]]',$$

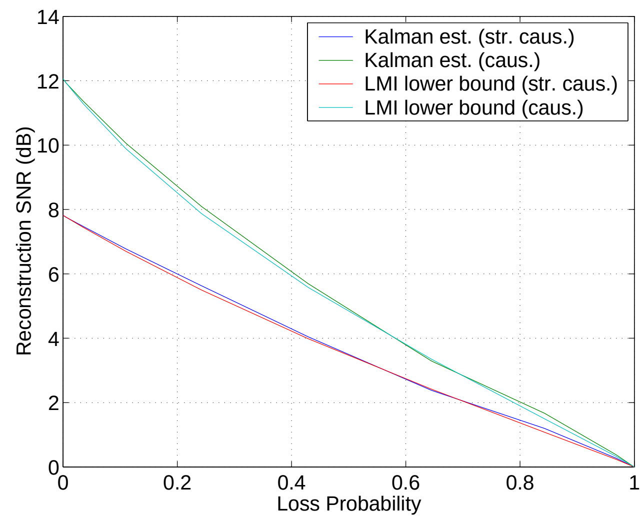

Figure 4.4. Reconstruction SNR of a lowpass signal from noisy samples with length 3 burst losses occuring with varying probabilities. Plotted are the strictly causal and causal estimates for the optimal Kalman filter, and the filter designed with the LMI method.

where the components $z_i[k]$ are independent, Gaussian, lowpass signals distributed identically to the signal $z[k]$ in the previous example. Suppose that at each time, k, we have a measurement of the form,

$$y_0[k] = z_1[k]\cos(\phi[k]) + z_2[k]\sin(\phi[k]) + d[k].$$

The observation represents a projection of $z[k]$ onto the line of angle $\phi[k]$ with additive noise $d[k]$. As before, we will assume that the noise power is 10 dB below the signal power of each component $z_i[k]$.

To make the problem interesting, we will assume that the observation angle, $\phi[k]$, is itself a Markov chain that can take on one of J values, $\phi[k] \in \{\phi_0, \ldots, \phi_{J-1}\}$, uniformly distributed on the unit circle,

$$\phi_j = \pi j/J.$$

We will assume the transition probabilities of $\phi[k]$ take the form

$$\mathbf{Pr}\left(\phi[k+1] = \phi_j \mid \phi[k] = \phi_i\right) = \begin{cases} 1-\mu, & \text{if } j = i; \\ \mu, & \text{if } j = i+1 \bmod J; \\ 0, & \text{otherwise,} \end{cases}$$

for some parameter $\mu \in [0,1]$. Thus, the observation angle, $\phi[k]$, "sweeps" through the J angles, advancing to the next angle with probability μ, and staying at the same angle with probability $1-\mu$. Finally, we will assume that the projection observations $y[k]$ are lost with

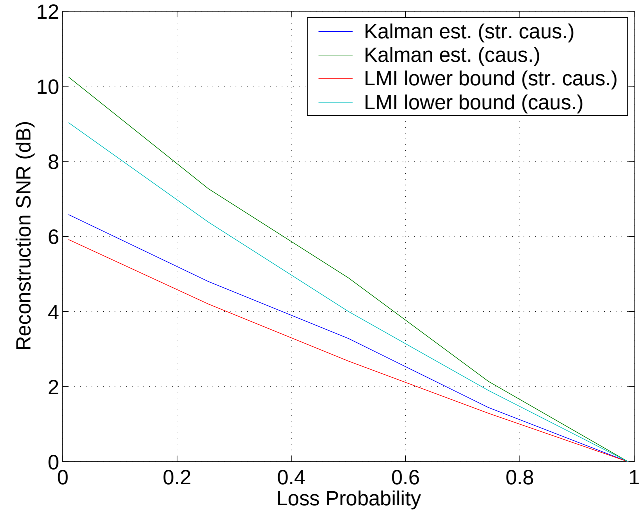

Figure 4.5. Reconstruction SNR of a two-dimensional lowpass signal with lossy random one-dimensional projections. Plotted are the strictly causal and causal estimates for the optimal Kalman filter, and the filter designed with the LMI method.

i.i.d. erasures with some probability λ. In this simulation, we take $J = 4$, $\mu = 0.85$ and vary the loss probability λ from 0 to 1.

The resulting system can be modeled easily as jump linear system. Modeling each component $z_i[k]$ with the two state low pass filter in the previous example, the two-dimensional signal $z[k]$ requires a total of four states. For the Markov process, observe that the i.i.d. erasures can be described with two states. Combining the two state erasures with the J-state observation angle, $\theta[k]$, we obtain a total of $2J$ Markov states - J states with erasures and J states without erasures. For the J non-erasure states, the observation matrices (C_{i2}, D_{i2}) are described by the observation angle, and for the J erased states, the matrices (C_{i2}, D_{i2}) are set to zero.

A comparison of the reconstruction SNR for the LMI estimation and optimal Kalman filtering is shown in Figure 4.5. As before, we compare both the strictly causal and causal estimates. The Kalman filter performance is based on a simulation of 4000 samples. Unlike the previous example, we see that there is a gap between the optimal Kalman filter and LMI-based estimator. Nevertheless, the gap is not large – approximately 1dB for the causal estimator and 0.5 dB for the strictly causal estimator.

While this two-dimensional tracking example is completely contrived, the point here is to illustrate that the jump linear system method is general and can incorporate a wide range of models. This specific example demonstrates that the jump linear estimation can incorporate non-independent Markov processes and multidimensional estimation. For such cases, the LMI estimator method can quickly compute a decent bound on the performance of the optimal Kalman estimator and provide a simple estimator to obtain that bound.

Chapter 5

State Space Design for Predictive Quantization

Predictive quantization is one of the simplest and most widely-used methods for encoding time-varying signals. Essentially, predictive quantization attempts to quantize changes in the signal from one sample to the next, rather than quantizing the samples themselves. If the signal is slowly varying relative to the sample rate, predictive quantization can result in a significant reduction in distortion for the same number of bits per sample. Due to its effectiveness and relative simplicity, predictive quantization is the basis of almost all speech encoders [42]. It is also used in some audio coders [63] and, in a sense, in motion compensated video coding [100].

Mathematically, a predictive quantizer encoder can be realized as a scalar quantizer in feedback with a linear filter called the *prediction filter*. The prediction filter attempts to "subtract out" information from the previous quantizer outputs. Its design depends on the source signal statistics. Since scalar quantization is a nonlinear operation, an exact analysis of the feedback system is difficult. In this chapter and the two subsequent chapters, an additive white noise model is used to represent the effect of quantization. Under this model, a predictive quantization system for a stationary source with rational power spectrum becomes a standard linear dynamical system. While the quantizer model has obvious limitations, it is important to emphasize that it is an eminently useful tool for analysis and design of systems that use only linear processing.

Of course, predictive quantization is a classical technique and has been extensively analyzed already. We will review some of the history in Section 5.1.2. The traditional textbook analysis of predictive quantization is performed in the frequency domain; it is computationally simple and can be easily related to the spectral characteristics of the source. See, for example, any standard textbook such as [64]. However, in order to connect predictive quantization with the Kalman filtering and linear systems techniques described earlier, this chapter reconsiders predictive quantization in state space. The results from this analysis are largely not new *per se*, but rather restatements of known results in state space form. In this sense, this chapter mostly serves as background for Chapters 6 and 7, where we combine the state space analysis of predictive quantization with the jump linear machinery of Chapter 3 to study the effect of losses.

The main result in this chapter is Theorem 5.1, which provides simple state space formulae for the optimal encoder filter design and the resulting reconstruction error. The optimal encoder transfer function and minimum error, of course, can also be derived in the frequency domain via a Wiener filter [25]. However, the state space formulae have the benefit of being able to easily incorporate the effect of the closed-loop quantization noise. Also, the state space solution shows an interesting connection between the effect of quantization and the effect of losses.

One benefit of the standard frequency-domain approach is that the predictive quantizer performance is easy to relate to rate–distortion theory. For example, based on the power spectral density of the quantization error, it is well-known that, at high rates, predictive quantization asymptotically achieves within a small constant factor the rate–distortion bound for a Gaussian random process [119, 5]. To illustrate how to relate the state space formulae to frequency-domain expressions, Section 5.8 rederives this asymptotic optimality result in state space. The section also briefly considers the rate–distortion performance of predictive quantization at low rates, where a sample rate optimization is proposed. This method is contrasted against pre-filtering techniques in [93], and more recently, [78].

This chapter is organized as follows: Predictive quantization is introduced in Section 5.1. That section includes a summary of the history of predictive quantization. Then, an additive white noise model for quantization noise is given in Section 5.2. Using the noise model, predictive quantization is described in state space in Section 5.3. Section 5.5 then gives the Kalman filter solution to the optimal design of the encoder and decoder filters. The final sections give a numerical example (Section 5.7) and relate the performance of predictive quantization to the rate–distortion bound for stationary Gaussian sources (Section 5.8).

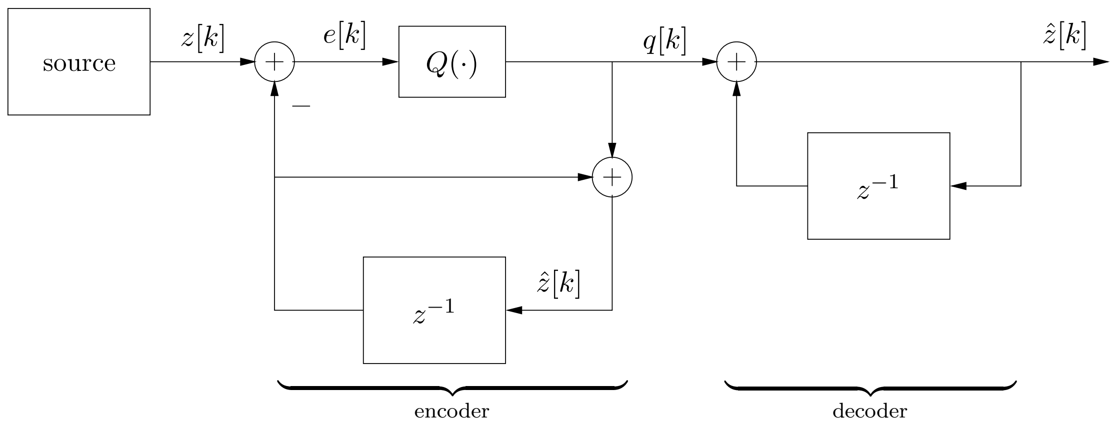

Figure 5.1. Differential quantizer encoder and decoder.

5.1 Predictive Quantization

The simplest predictive quantizer is the differential quantizer shown in Figure 5.1. The signal to be quantized is denoted $z[k]$, and in Figure 5.1 we have explicitly depicted both the encoder and the decoder. The encoder takes the source signal $z[k]$ and produces a sequence of quantizer outputs $q[k]$, from which the decoder produces the signal estimate $\hat{z}[k]$. The differential quantizer encoder is a scalar quantizer Q in feedback with an integrator. It is described by the recursive equations

$$\begin{aligned} q[k] &= Q(z[k] - \hat{z}[k-1]), \\ \hat{z}[k] &= \hat{z}[k-1] + q[k]. \end{aligned} \tag{5.1}$$

At each time step, the encoder quantizes the difference between the current sample $z[k]$ and the previous signal estimate $\hat{z}[k-1]$ to yield the quantized output $q[k]$. The samples $q[k]$ are integrated to produce the signal estimate $\hat{z}[k]$. The decoder also integrates the samples $q[k]$ to reproduce the estimate $\hat{z}[k]$.

The intuition for the differential quantizer is simple: In (5.1) the quantizer quantizes only the difference between the consecutive samples as opposed to the samples themselves. If $z[k]$ is positively correlated with $z[k-1]$ (for example, the signal is slowly varying), then the differences have a smaller variance than the signal itself. The quantizer can thus achieve a greater accuracy for the same number of bits.

The differential quantizer can be extended to a more general predictive quantizer shown in Figure 5.2. The integrators in the encoder and decoder are replaced by general filters denoted H_{enc} and H_{dec}. Throughout, we will assume that both the encoder and decoder

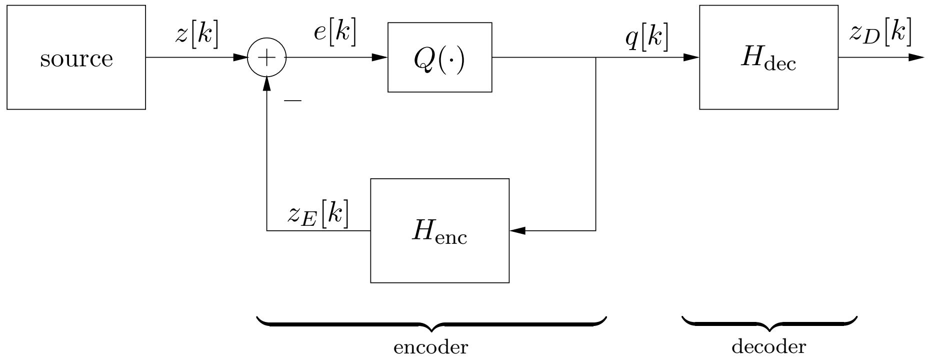

Figure 5.2. Predictive quantizer encoder and decoder with a general higher-order filter.

filters are causal, linear and time invariant. In addition, we will assume that the encoder filter is strictly causal, meaning that its output at time k depends only on inputs up to sample $k-1$. The output of the encoder filter is denoted $z_E[k]$ and the output of the decoder is denoted $z_D[k]$. The decoder output $z_D[k]$ is the final estimate of the original signal $z[k]$, so ideally $z_D[k]$ is close to $z[k]$.

To place the differential quantizer system (5.1) into the format of Figure 5.2, we simply define the encoder and decoder filters by

$$H_{\text{enc}}: \quad z_E[k+1] = z_E[k] + q[k],$$

and

$$H_{\text{dec}}: \quad z_D[k] = z_D[k-1] + q[k].$$

In z domain we have

$$H_{\text{enc}}(z) = \frac{z^{-1}}{1-z^{-1}} \qquad \text{and} \qquad H_{\text{dec}}(z) = \frac{1}{1-z^{-1}}.$$

Observe that encoder filter H_{enc} is strictly causal and the decoder filter H_{dec} is causal. If initially $z_E[1] = z_D[0]$, then it can be verified that $z_E[k+1] = z_D[k]$ for all k. In this case, if we define $\hat{z}[k] = z_D[k] = z_E[k+1]$, then $\hat{z}[k]$ will satisfy (5.1).

5.1.1 Autoregressive Sources and Prediction

Let

$$z[k] = \sum_{i=1}^{K} \alpha_i z[k-i] + w[k] \qquad \text{for all } k \in \mathbb{Z}, \tag{5.2}$$

where $w[k]$ is a zero-mean white random process with variance σ_w^2 and $w[k]$ is uncorrelated with $\{z[k-i]\}_{i\in\mathbb{Z}^+}$. If the roots of the polynomial $1-\sum_{i=1}^{K}\alpha_i t^k$ are all inside the unit circle, then $z[k]$ is a wide-sense stationary random process and is called an *autoregressive (AR) process of order* K. In the case that $w[k]$ is Gaussian, $z[k]$ is also Gaussian. Autoregressive sources are good for understanding predictive quantization, and in the Gaussian case the rate–distortion function of the source can be easily computed. This is explored further in Section 5.8. Of particular interest are first-order AR processes, denoted AR-1. For AR-1 processes we omit the subscript on α_1.

The power of the AR source defined above is at least as large as the power of $w[k]$. The conditional expectation

$$E[z[k] \mid z[k-1], z[k-2], \ldots] = \sum_{i=1}^{K} \alpha_i z[k-i]$$

implies that if $\hat{z}[k] = \sum_{i=1}^{K} a_i z[k-i]$, then

$$\min_{a_1,a_2,\ldots,a_K} E[(z[k]-\hat{z}[k])^2] = \sigma_w^2, \qquad \text{achieved with } a_i = \alpha_i \ \forall i.$$

So, neglecting for the moment the fact that the input to the encoder filter is quantized, the power of the quantizer input is minimized by

$$H_{\text{enc}}(z) = \frac{H_{\text{pred}}(z)}{1 - H_{\text{pred}}(z)}$$

where

$$H_{\text{pred}}(z) = \sum_{i=1}^{K} a_i z^{-i}$$

is the optimal strictly causal filter for predicting $z[k]$ from its own past samples. These filters are called *matched* to the source.

5.1.2 History and Prior Analyses

We are employing here the term "predictive quantization" as used, for example, by Gersho and Gray [64] and Gray and Neuhoff [75]. This is to avoid the archaic term *differential pulse code modulation (DPCM)*, which should be reserved for situations involving the digital representation of analog information followed by modulation.

According to the comprehensive tutorial history of quantization by Gray and Neuhoff [75], the idea of predictive quantization originated in a 1946 French patent of Derjavitch, Deloraine, and Van Mierlo, though a US patent was granted to Cutler in 1952.

Other references from the early 1950s include papers by DeJager; Harrison; Kretzmer; Oliver, Pierce, and Shannon; and Zetterberg; see [75] for full bibliographic details. The Ph.D. dissertation of Elias and the resulting papers [51, 52] explained and to some extent popularized predictive coding, but they do not actually analyze effects of quantization at all; rather, the focus is on connecting Wiener's prediction theory with Shannon's rate–distortion theory.

Analysis of quantization tends to follow one of two paths: (a) derivation of necessary and sufficient conditions for optimality; and (b) high-rate analysis that depends on approximating the probability density of the quantizer input with a piecewise constant function. Feedback causes significant complications, to the point where a purist would say that very little has been rigorously established for predictive quantization. Conclusive results are subject to at least one of the following limitations: Gaussian AR-1 sources, matched predictors, and one-bit quantizers. But at the same time, it has been demonstrated that matched predictors are not optimal, and clearly results for more general sources and quantizers are desired.

One line of analysis for predictive quantization was started by Gish [66], Fine [57], Davisson [43], and O'Neal [118] and continued by Arnstein [8]. It is focused on exact performance computation but limited to the case of coding a Gaussian AR source as in (5.2) with a matched predictor. The match between the predictor and the generating model for $z[k]$ allows one to write a generating model for the quantizer input as

$$e[k] = w[k] + \sum_{i=1}^{K} a_i(e[k-i] - Q(e[k-i])).$$

In this form it is clear that $e[k]$ is a Kth order Markov process. However, because of the quantization $e[k]$ is not a Gaussian process, and it is difficult to find the distribution $e[k]$. Arnstein [8] uses an orthogonal expansion with respect to Hermite polynomials to approximate the density of $e[k]$. This computation is complicated even for a low order K and for a two-level quantizer Q; it is infeasible to use it rigorously in a design procedure for Q or H_{enc}.

Subsequent attempts at precise analysis are predominantly limited further—not just to the Gaussian AR case, but to AR-1. Janardhanan [84] extended the analysis of Arnstein [8] to quantizers with more levels and resolved certain convergence issues. Farvardin and Modestino [56] used similar techniques to study the performance of predictive quantization with entropy-constrained quantization, and Gerr and Cambanis [61] investigated performance with adaptation of the step size of a symmetric uniform quantizer. Stability of the adaptive case is also studied by Kieffer [92].

While the use of a matched predictor simplifies the analysis and is intuitively a reasonable choice, it is not generally optimal. Exact performance analysis for a Gaussian AR-1 source coded with a mismatched predictor is given in [116]. It is demonstrated there that the optimum predictor is not matched, and a nonrigorous quantizer optimization technique is proposed. The use of entropy-constrained quantization and assuming high rate simplifies the situation analytically because the optimum quantizer is uniform [67]. Under these assumptions, a matched predictor is optimal and results in performance 1.53 dB worse than the rate–distortion bound [119, 5].

Faced with the existing literature, the question of how to analyze and design predictive quantization systems becomes essentially philosophical: either become caught up in technicalities that are necessary for making precise, rigorous statements regarding the optimality of a particular quantizer choice and its performance, or else make vast simplifications at the peril of rigor. This dissertation takes the latter path, but not lightly. In all the existing literature, there is no evidence that the accuracy of the *quantizer* analysis has much impact on the optimal choice of the *filters* in the system. The following section describes a simple quantizer model and its limitations. This model is used throughout the remainder of the dissertation.

5.2 Additive White Noise Quantizer Model

As stated earlier, due to the nonlinear nature of the scalar quantizer, an exact analysis of the predictive quantizer system is difficult. The classical approach to deal with the nonlinearity is to approximate the quantizer as a linear gain with additive white noise (AWN). One such model is provided by the following Lemma.

Lemma 5.1 *Consider the predictive quantizer system in Figure 5.2. Suppose that the closed-loop system is well-posed and stable and all the resulting signals are wide sense stationary random processes with zero mean. In addition, suppose that the scalar quantizer function $Q(\cdot)$ is designed such that for any input e,*

$$\mathbf{E}\left(\, e \mid Q(e) \, \right) = Q(e). \tag{5.3}$$

Also, let

$$\beta = \frac{\mathbf{E}|e[k] - q[k]|^2}{\mathbf{E}|e[k]|^2}. \tag{5.4}$$

Then, the quantizer output can be written

$$q[k] = \rho e[k] + \sigma_q w_q[k] \tag{5.5}$$

where $w[k]$ *is a zero-mean, unit variance signal, uncorrelated with* $e[k]$, *and*

$$\begin{aligned} \rho &= 1 - \beta \\ \sigma_q^2 &= \beta(1-\beta)\mathbf{E}|e[k]|^2. \end{aligned} \tag{5.6}$$

The proof of this lemma is given at the end of the section, and is typical for linear analyses with AWN quantizer models. The result shows that the quantizer can be described as a linear gain ρ, along with additive noise $w_q[k]$ uncorrelated with the quantizer input $e[k]$. Observe that since $w_q[k]$ is normalized to have unit variance, the variance of the additive noise is σ_q^2.

The assumption (5.3) is that, for each partition region, the function Q will output the conditional mean. As discussed in Section 2.8.3, this condition will be satisfied for any optimally designed Lloyd-Max quantizer. Also, even if the quantizer is not optimal, such as a uniform quantizer, the quantizer output values can be modified to satisfy (5.3). Moreover, the condition will also approximately hold for uniform quantizers without modification of the output values used at high rates.

The factor β in (5.4) is a proportionality constant between the quantizer input variance and quantizer error variance. The parameter can thus be seen as a measure of the quantizer's relative accuracy. In Gersho and Gray [64], the factor $1/\beta$ is called the *coding gain* of the quantizer. Following this terminology, we will call β either the *inverse coding gain* or *coding loss*. The coding gain is often quoted in dB:

$$\text{coding gain (dB)} = 10\log_{10}(1/\beta) = -10\log_{10}\beta.$$

In general, β will depend on the quantizer design and number of bits per sample. In Section 2.8.1, the coding gain was called the reconstruction SNR. The section describes the coding gains for several simple quantizers of a Gaussian source.

While Lemma 5.1 provides a simple additive noise model for the quantizer, we require two further approximations for the linear modeling:

(a) At each time sample k, the quantizer noise $w_q[k]$ constructed in Lemma 5.1 is guaranteed to be uncorrelated with the quantizer input $e[k]$. However, in the linear modeling, we need to further assume that $w_q[k]$ is white and uncorrelated with *all* samples $e[j]$ up to time $j = k$.

(b) In general, the factor β depends on the particular quantizer design along with the probability distribution of the quantizer input $e[k]$. However, the quantizer input $e[k]$ will typically have a complex distribution due to the nonlinear nature of the quantizer that is in feedback with the encoder filter. This distribution is difficult to characterize. The second approximation we need is that, for the purpose of the estimating the coding loss, the quantizer input can be treated as Gaussian.

Neither of these two assumptions is exactly valid; they instead represent approximations. Nevertheless, the assumptions are widely used and have been proven to be very accurate, particularly at high rate. Moreover, recent theoretical analyses, such as [117], have given some theoretical justification to the white, Gaussian assumption. At the end of the chapter, we will see that the error in the assumptions is negligible and the linear modeling of the quantizer provides an extremely accurate model for predicting the predictive quantization performance.

Proof of Lemma 5.1: To simplify the notation in this proof, we will omit the dependence on the time sample k on all signals. For example, we will write q for $q[k]$.

Now, from (5.3), $\mathbf{E}(qe \mid q) = q^2$, so

$$\mathbf{E}(qe) = \mathbf{E}(q^2). \tag{5.7}$$

Combining (5.7) and (5.4), we obtain

$$\begin{aligned}\beta\mathbf{E}e^2 &= \mathbf{E}(q-e)^2 = \mathbf{E}q^2 - 2\mathbf{E}qe + \mathbf{E}e^2 \\ &= -\mathbf{E}q^2 + \mathbf{E}e^2.\end{aligned}$$

Therefore,

$$\mathbf{E}q^2 = (1-\beta)\mathbf{E}e^2. \tag{5.8}$$

Now, let $v = q - (1-\beta)e$, so that

$$q = (1-\beta)e + v.$$

Since q and e are zero mean, so is v. Also, using (5.7) and (5.8),

$$\mathbf{E}(ve) = \mathbf{E}(q - (1-\beta)e)e) = (1-\beta)\mathbf{E}(e^2) - (1-\beta)\mathbf{E}(e^2) = 0,$$

so v and e are uncorrelated. Finally,

$$\begin{aligned}\mathbf{E}(v^2) &= \mathbf{E}(q-(1-\beta)e)^2 \\ &= \mathbf{E}(q^2) - 2(1-\beta)\mathbf{E}(qe) + (1-\beta)^2\mathbf{E}(e^2) \\ &= \left[(1-\beta) - 2(1-\beta)^2 + (1-\beta)^2\right]\mathbf{E}(e^2) \\ &= \beta(1-\beta)\mathbf{E}(e^2).\end{aligned}$$

Therefore, if we define σ_q and ρ as in (5.6) and let

$$w_q = (1/\sigma_q)v,$$

we have that w_q is a zero-mean process, uncorrelated with e, with unit variance and satisfying,

$$q = \rho e + \sigma_q w_q.$$

This completes the proof. □

5.3 Linear State Space Predictive Quantizer System Model

Substituting the linear quantizer model in (5.5) into the predictive quantizer system in Figure 5.2, one arrives at a linear system depicted in Figure 5.3. As stated in the previous section, we will assume that the quantizer noise signal, $w_q[k]$, is a zero-mean signal, uncorrelated with the quantizer input, $e[k]$.

Also shown in Figure 5.3, is a signal model for the source $z[k]$. Specifically, $z[k]$ is described as filtered white noise. This model is similar to the model discussed in Section 2.5. The filter, H_z, can be selected to match the statistics of the source signal process, $z[k]$. We will assume the filter, H_z can be represented in state space form,

$$\begin{aligned}x[k+1] &= Ax[k] + Bw_z[k] \\ z[k] &= Cx[k] + Dw_z[k]\end{aligned} \tag{5.9}$$

where $w_z[k]$ is zero-mean, white noise with unit variance. We will assume the filter H_z is stable with $DD' > 0$.

Using the AWN quantizer model and linear system signal description, we have reduced the predictive quantizer system to a linear system with white noise inputs. This system can now analyzed with Kalman filtering methods discussed earlier.

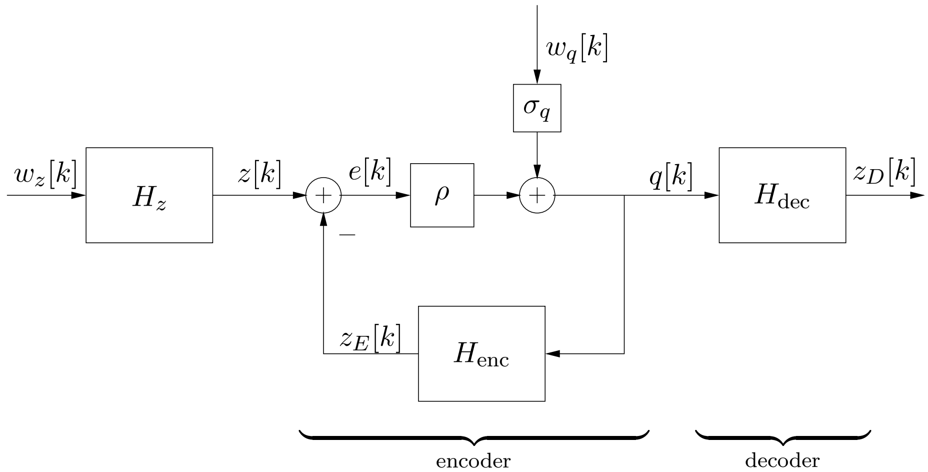

Figure 5.3. Linear model for the predictive quantizer system. The quantizer is replaced by a linear gain and additive noise, and the source signal, $z[k]$, is described as filtered white noise.

5.4 Encoder and Decoder Filter Design Problem

Designing the quantizer system for a given source $z[k]$, amounts to selecting the encoder and decoder filters, $H_{\rm enc}$ and $H_{\rm dec}$. For the linear system model, it is also necessary to determine the quantization error variance σ_q^2 that is consistent with the quantizer's coding loss, β. The constraints for this design are specified in the following definition.

Definition 5.1 *Consider the linear predictive filter quantizer in Figure 5.3. Fix a stable linear, time-invariant source filter H_z and quantization coding loss $\beta \in [0, 1]$. For this model, an encoder-decoder design pair $(H_{\rm enc}, H_{\rm dec})$ will be called* feasible *if*

(a) The encoder and decoder filters, $H_{\rm enc}$ and $H_{\rm dec}$, are linear, time-invariant filters with $H_{\rm dec}$ causal and $H_{\rm enc}$ strictly causal;

(b) The resulting closed-loop map $(w_z, w_q) \mapsto (z, z_E, z_D)$ is well-posed and stable; and

(c) There exists a quantization noise level σ_q^2 such that when the inputs $w_z[k]$ and $w_q[k]$ are zero-mean, unit variance white noise and $\rho = 1 - \beta$, the closed-loop quantizer input signal $z[k] - z_E[k]$ satisfies

$$\sigma_q^2 = \beta(1-\beta) \lim_{k\to\infty} \mathbf{E}|z[k] - z_E[k]|^2. \tag{5.10}$$

The final constraint (c) requires some explanation. In general, the quantization noise variance is not only a function of the quantizer itself, but also the quantizer input variance that results with the choice of the encoder, H_{enc}. This is a consequence of the fact that the quantization noise scales with the quantizer input variance by the factor β. In Definition 5.1, the quantization noise level σ_q^2 is thus defined as the level that is consistent with the closed-loop encoder system and the linear quantizer model in Lemma 5.1. We will call the quantization noise level, σ_q^2 in part (c), the *closed-loop quantization noise variance for the encoder* H_{enc}.

Now, given a feasible design, we define the asymptotic mean-squared error (MSE) of the reconstruction as

$$\text{MSE} = \lim_{k\to\infty} \mathbf{E}|z[k] - z_D[k]|^2, \tag{5.11}$$

where $z_D[k]$ is the decoder output, and the encoder-decoder system is assumed to operate with the consistent closed-loop quantization noise variance. The limit here is a technical point used to avoid the effect of initial conditions. Since our definition of a feasible design includes only time-invariant filters resulting in a stable closed-loop system, the limit will always exist. We will say that a feasible design is *optimal* if it minimizes the asymptotic MSE.

Under the linear AWN model for the quantizer, the optimal encoder and decoder filters can be described as certain optimal estimators. To see this, first consider the decoder, H_{dec}. Since the input to the decoder H_{dec} is the quantizer sequence $q[k]$, the optimal choice for H_{dec} must be the causal MMSE estimate of $z[k]$ given $q[k]$. That is, the optimal decoder output must be

$$z_D[k] = \hat{z}[k \mid k] = \mathbf{E}\left(z[k] \mid q[0:k]\ \right),$$

where we have used the notation from Section 2.4, where $\hat{z}[k \mid j]$ denotes the MMSE estimate of $z[k]$ given the observed data $q[0:j]$.

Now consider the encoder filter. Observe that, given $q[k]$, the output of the encoder filter, H_{enc}, is completely known to the decoder. Therefore, aside from the quantization variance σ_q^2, the output of the encoder filter has no bearing on the achievable MSE at the decoder. Consequently, to minimize the MSE at the decoder, the encoder filter H_{enc}, should be selected strictly to minimize the quantization noise variance σ_q^2. But, for a fixed coding loss β, (5.10) shows that σ_q^2 is minimized by minimizing the quantizer input variance, $\mathbf{E}|e[k]|^2$, where $e[k] = z[k] - z_E[k]$. Since H_{enc} must be strictly causal, the quantizer input variance is minimized by setting $z_E[k] = \hat{z}[k \mid k-1]$, the MMSE strictly causal estimate of

$z[k]$. That is, the optimal encoder filter output must be

$$z_E[k] = \hat{z}[k \mid k-1] = \mathbf{E}\left(z[k] \mid q[0:k-1]\ \right).$$

This observation explains the term *predictive encoding*: the encoder finds the best prediction of the current sample $z[k]$ given the previous quantized outputs. The prediction is subtracted out and the quantizer is applied to the prediction error.

These observations are summarized in the following proposition.

Proposition 5.1 *Consider the linear predictive filter quantizer in Figure 5.3, and fix a stable linear-time invariant source filter H_z and quantization coding loss $\beta \in [0,1]$. If an encoder-decoder pair $(H_{\rm enc}, H_{\rm dec})$ is optimal, then*

(a) The output of the encoder filter, $H_{\rm enc}$, is the strictly causal MMSE estimate of $z[k]$ given $q[k]$,

$$z_E[k] = \hat{z}[k|k-1] = \mathbf{E}\left(z[k]\ \mid q[0:k-1]\ \right).$$

(b) The output of the decoder filter, $H_{\rm dec}$, is the causal MMSE estimate of $z[k]$ given $q[k]$,

$$z_D[k] = \hat{z}[k|k] = \mathbf{E}\left(z[k]\ \mid q[0:k]\ \right).$$

This proposition is essentially a restatement of what Gersho and Gray [64] call the *Fundamental Theorem of Predictive Quantization.*

5.5 Kalman Filter Solution

Proposition 5.1 shows that the optimal encoder and decoder filters are simply MMSE estimators of a source signal $z[k]$ from the quantizer output signal $q[k]$. With the linear system model in Section 5.3, the equations for these estimators are given by the Kalman filter equations. The basic results are summarized in the following theorem, whose proof is given at the end of the section.

Theorem 5.1 *Consider the linear predictive filter quantizer in Figure 5.3, and fix a stable, linear, time-invariant source filter H_z as in (5.9) and quantization coding loss $\beta \in [0,1]$.*

Then one optimal encoder filter, H_{enc}, *can be described by the state space equations,*

$$\begin{aligned} x_E[k+1] &= Ax_E[k] + Lq[k] \\ z_E[k] &= Cx_E[k] \end{aligned} \tag{5.12}$$

where the gain matrix L *is given by*

$$L = E_1 G \tag{5.13}$$

and E_1 *and* G *are derived from the positive semidefinite solution* $P \geq 0$ *to the* modified algebraic Riccati equation

$$\begin{aligned} P &= APA' - (1-\beta)E_1 G E_1' + BB' \\ E_1 &= APC' + BD' \\ G &= (CPC' + DD')^{-1} \end{aligned} \tag{5.14}$$

The optimal decoder filter is given by the state space equations,

$$\begin{aligned} x_D[k+1] &= Ax_D[k] + Lq[k] \\ z_D[k] &= Cx_D[k] + q[k] \end{aligned} \tag{5.15}$$

Finally, the closed-loop quantization noise variance σ_q^2 *and the asymptotic MSE in (5.11) are given by*

$$\begin{aligned} \sigma_q^2 &= \beta(1-\beta)(CPC' + DD') \\ MSE &= \beta(CPC' + DD'). \end{aligned}$$

The theorem provides simple state space formulae for the optimal encoder and decoder filters. The state space matrices contain the A and C matrices of the source signal state space system (5.9), along with a gain matrix L derived from the solution to a modified algebraic Riccati equation (5.14).

There are two interesting points to note in this result. First, the state space equations for the encoder and decoder filters in (5.12) and (5.15), respectively, differ only by a factor $q[k]$ in the output equation. Therefore, their transfer functions are related by the simple identity,

$$H_{\text{dec}}(z) = H_{\text{enc}}(z) + 1.$$

Moreover, the state update equations in the encoder and decoder equations are identical. Consequently, if the systems have the same initial conditions (i.e. $x_E[0] = x_D[0]$) then,

$x_E[k] = x_D[k]$ for all subsequent samples k. Thus, the encoder and decoder will have the state estimates. In the next chapter, we will see that, in the presence of channel losses, this property no longer holds: With losses of the quantizer output, the decoder will not, in general, know the encoder state.

A second point to note is that the modified algebraic Riccati equation in (5.14) is precisely the modified equation in (4.18) in Section 4.5. The latter equation arose in the context of the lossy observations. We thus see an interesting relationship between quantization noise and loss: *the effect of quantization with coding loss β is equivalent to estimating a signal with independent losses with a loss probability of β.* Moreover, the modified algebraic Riccati equation in (5.14) can be solved with the same iterative procedure in Section 4.5 for a simple way to obtain the optimal prediction filter and decoder.

Proof of Theorem 5.1: Combining (5.9) with (5.5),

$$\begin{aligned} x[k+1] &= Ax[k] + Bw_z[k] \\ z[k] &= Cx[k] + Dw_z[k] \\ q[k] &= \rho Cx[k] + \rho Dw_z[k] + \sigma_q w_q[k] + u[k] \end{aligned} \tag{5.16}$$

where

$$u[k] = -\rho z_E[k].$$

If we define the vector noise $w[k] = [w_z[k] \; w_q[k]]$, and let

$$\begin{aligned} B_1 &= [B \; 0], \\ C_1 &= C, \quad C_2 = \rho C, \\ D_1 &= [D \; 0], \quad D_2 = [\rho D \; \sigma], \end{aligned}$$

then (5.16) can be rewritten

$$\begin{aligned} x[k+1] &= Ax[k] + B_1 w[k] \\ z[k] &= C_1 x[k] + D_1 w[k] \\ q[k] &= C_2 x[k] + D_2 w[k] + u[k]. \end{aligned} \tag{5.17}$$

Now, Proposition 5.1 shows that the output of the optimal encoder filter H_{enc} is given by $z_E[k] = \hat{z}[k|k-1]$, the MMSE estimate of $z[k]$ given the past samples $q[k-1], q[k-2], \ldots$. Using the state space model (5.17), this estimate is given precisely by the Kalman filter in Section 2.4. Since we are concerned only with asymptotic MSE, we can use the steady-state Kalman filter equations in (2.18),

$$\begin{aligned} \hat{x}[k+1|k] &= A\hat{x}[k|k-1] + L(q[k] - C_2\hat{x}[k|k-1] - u[k]) \\ \hat{z}[k|k-1] &= C_1\hat{x}[k|k-1], \end{aligned}$$

where L is the Kalman gain matrix. Note that we have used the fact that $u[k]$ is a known input, since it can be computed from $z_E[k] = \hat{z}[k|k-1]$. If we define the state $x_E[k] = \hat{x}[k|k-1]$, and use the fact that $z_E[k] = \hat{z}[k|k-1]$, the encoder equations can be re-written as

$$\begin{aligned} x_E[k] &= Ax_E[k] + L(q[k] - C_2 x_E[k] - u[k]) \\ z_E[k] &= C_1 x_E[k], \end{aligned} \tag{5.18}$$

Now, since $C_2 = \rho C$,

$$u[k] = -\rho z_E[k] = -\rho C\hat{x}[k|k-1] = -C_2\hat{x}[k|k-1]. \tag{5.19}$$

Applying this identity along with the fact that $C_1 = C$ into (5.18), we obtain the encoder equations in (5.12).

Next, we derive the expression for the Kalman gain matrix L using Theorem 2.1. To this end, first observe that, using Theorem 2.1, the quantizer input variance is given by

$$\mathbf{E}\|e[k]\|^2 = \mathbf{E}\|z[k] - \hat{z}[k|k-1]\|^2 = CPC' + DD',$$

where P is the asymptotic error variance matrix. Using (5.10), the quantizer error variance is given by

$$\sigma_q^2 = \beta(1-\beta)\mathbf{E}\|e[k]\|^2 = \beta(1-\beta)(CPC' + DD').$$

Now using the fact that $C_2 = \rho C$ and $D_2 = [\rho D \ \sigma_q]$, the matrix G in (2.17) is given by

$$\begin{aligned} G &= (C_2PC_2' + D_2D_2')^{-1} = (\rho^2 CPC' + \rho^2 DD' + \sigma_q^2)^{-1} \\ &= ((1-\beta)^2(CPC' + DD') + \beta(1-\beta)(CPC' + DD'))^{-1} \\ &= \frac{1}{1-\beta}(CPC' + DD')^{-1}. \end{aligned} \tag{5.20}$$

Thus,

$$\begin{aligned} P &= APA' + (APC_2' + B_1D_2')G(C_2PA' + D_2B_1') + B_1B_1' \\ &= APA' + \frac{\rho^2}{1-\beta}APC'(CPC' + DD')^{-1}CPA' + BB' \\ &= APA' + (1-\beta)(APC' + BD')(CPC' + DD')^{-1}(CPA' + DB') + BB' \\ &= APA' + (1-\beta)E_1GE_1' + BB', \end{aligned}$$

where

$$E_1 = APC' + BD', \quad G = (CPC' + DD')^{-1}.$$

This proves (5.14). Substituting (5.20) into the expression for L in Theorem 2.1,

$$\begin{aligned} L &= (APC_2' + B_1D_2')G = \frac{\rho}{1-\beta}(APC' + BD')(CPC' + DD')^{-1} \\ &= (APC' + BD')(CPC' + DD')^{-1} = E_1G, \end{aligned}$$

which proves (5.13).

Now, substituting the expressions for C_1, C_2, D_1 and D_2 into the expression for E_2 in Theorem 2.1, we obtain

$$E_2 = C_1PC_2' + D_1D_2' = \rho(CPC' + DD'). \tag{5.21}$$

Substituting (5.19), (5.20) and (5.21) in (2.18),

$$\begin{aligned}
\hat{z}[k|k-1] &= C_1\hat{x}[k|k-1] = C\hat{x}[k|k-1] \\
\hat{z}[k|k] &= C_1\hat{x}[k|k-1] + EG(q[k] - C_2\hat{x}[k|k-1] - u[k]) \\
&= C\hat{x}[k|k-1] + q[k]
\end{aligned}$$

By Proposition 5.1, the optimal decoder output is given by $z_D[k] = \hat{z}[k|k]$. Therefore, if we define the decoder state as $x_D[k] = \hat{x}[k|k-1]$, we obtain the decoder equations,

$$\begin{aligned}
x_D[k] &= Ax_D[k] + Lq[k] \\
z_D[k] &= Cx_D[k] + q[k],
\end{aligned} \tag{5.22}$$

Finally, substituting (5.20) and (5.21) into the expression for the asymptotic MSE in Theorem 2.2,

$$\begin{aligned}
\text{MSE} &= \lim_{k\to\infty} \mathbf{E}|z[k] - z_D[k]|^2 \\
&= \lim_{k\to\infty} \mathbf{E}|z[k] - \hat{z}[k|k]|^2 \\
&= C_1PC_1' + D_1D_1' - E_2GE_2' \\
&= CPC' + DD' - \frac{\rho^2}{1-\beta}(CPC' + DD') \\
&= \beta(CPC' + DD').
\end{aligned}$$

This completes the proof. □

5.6 High Rate Limit and One-Step Ahead Prediction

A well-known property of predictive quantizers is that, at high rates, the optimal encoder filter is closely related to the optimal one-step ahead prediction filter. This property is easy to derive by comparing the state space formulae in Theorem 5.1 with the formulae for the one-step ahead prediction filter in Section 2.4.4.

To see the precise relationship, consider the limit of Theorem 5.1 as the coding loss β approaches zero; or, equivalently, the quantization rate is high. As $\beta \to 0$, the modified algebraic Riccati equation (5.14) reduces to the standard algebraic Riccati equation (2.21) for the one-step ahead prediction problem. Thus, the solution P to the modified Riccati equation (5.14) reduces to the asymptotic one-step ahead prediction error state variance, P, in (2.21). Moreover, the optimal encoder filter in (5.12) is identical to the one-step ahead prediction filter (2.20) with input $q[k]$ replaced by $z[k] - C\hat{x}[k|k-1] = z[k] - \hat{z}[k|k-1]$. Therefore, the transfer functions of the one-step ahead prediction and encoder filters are related by

$$H_{\text{pred}}(z) = H_{\text{enc}}(z)(1 - H_{\text{pred}}(z)),$$

from which we conclude the identity

$$H_{\text{enc}}(z) = \frac{H_{\text{pred}}(z)}{1 - H_{\text{pred}}(z)}.$$

Using the formula $H_{\text{enc}} = H_{\text{dec}} - 1$,

$$H_{\text{dec}}(z) = H_{\text{enc}}(z) + 1 = \frac{1}{1 - H_{\text{pred}}(z)}.$$

Thus, in the limit of high rates, the optimal encoder and decoder filters are described by simple transformations of the optimal one-step ahead predictor.

5.7 Numerical Example

To illustrate the state space encoder filter design and analysis, we consider a simple numerical example. The signal, $z[k]$, to be quantized is modeled as the filter output

$$z = H_{\text{LP}}(z)w_1 + \sigma_2 w_2,$$

where $H_{\text{LP}}(z)$ is a second order lowpass filter, and w_1 and w_2 are independent, unit variance, white noise random processes. The term $H_{\text{LP}}(z)w_1$ represents a dominant lowpass component of the signal, while the second term $\sigma_2 w_2$ models a small white noise component that is added to insure that $z[k]$ has non-zero power for all frequencies. In this experiment, the variance of $\sigma_2 w_2$ is set to 20 dB below the lowpass component $H_{\text{LP}}(z)w_1$.

To obtain a simple realization for the filter $H_{\text{LP}}(z)$, we use the Chebyshev filter design program, `cheby1`, in MATLAB with the passband set to 0.1 times the sample rate and a passband ripple of 0.5 dB. The resulting filter for z with the noise inputs (w_1, w_2), admits a state space realization of the form (5.9) with system matrices

$$\begin{aligned} A &= \begin{bmatrix} 0.582 & -0.309 \\ 0.309 & 0.940 \end{bmatrix}, \quad B = \begin{bmatrix} 0.345 & 0 \\ 0.069 & 0 \end{bmatrix}, \\ C &= [0.332 \;\; 2.08], \quad D = [0.075 \;\; 0.1] \end{aligned}$$

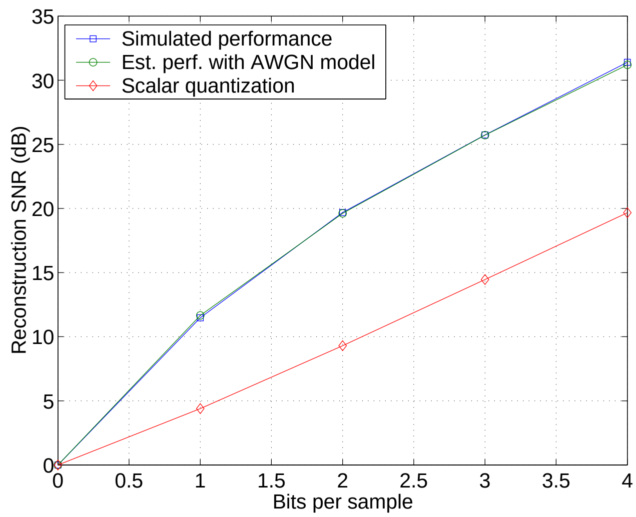

Figure 5.4. Estimation of predictive coding performance with linear AWN model. The simulated curve represents the performance measured in a 10000 sample simulation with a true uniform scalar quantizer. The predicted performance is based on the linear AWN quantizer model. The two curves almost overlap. Also plotted for comparison is the reconstruction SNR with scalar quantization. The signal model is a second-order lowpass Gaussian random process with bandwidth 0.1 times the sample rate.

Using this state space model for the signal $z[k]$, we designed predictive quantizer encoders and decoders using the state space formulae in Theorem 5.1. In this experiment, we used a uniform quantizer with 1 to 4 bits per sample with the step size selected assuming a Gaussian input. The coding gain, $10\log_{10}(1/\beta)$, for the uniform scalar quantizer is given in Figure 2.1.

The results of the quantizer design are shown in Figure 5.4. The curve labeled "simulated performance" represents the average performance from a 10000 random sample realization with the true scalar quantizer. The performance is measured by the reconstruction SNR as given by

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\mathbf{E}|z[k]|^2}{\mathbf{E}|z[k]-z_D[k]|^2}\right),$$

which represents the relative signal error. The reconstruction SNR is plotted as a function of the quantizer bit rate. Also plotted is the estimated reconstruction SNR given by the linear AWN quantizer model MSE in Theorem 5.1. The simulated and linear model estimate nearly overlap showing the AWN quantizer model is very accurate.

For comparison, Figure 5.4 also shows the performance of scalar quantization with no prediction. It can be seen that predictive quantization has a 7.2 to 11.5 dB *prediction gain* over scalar quantization depending on the bit rate.

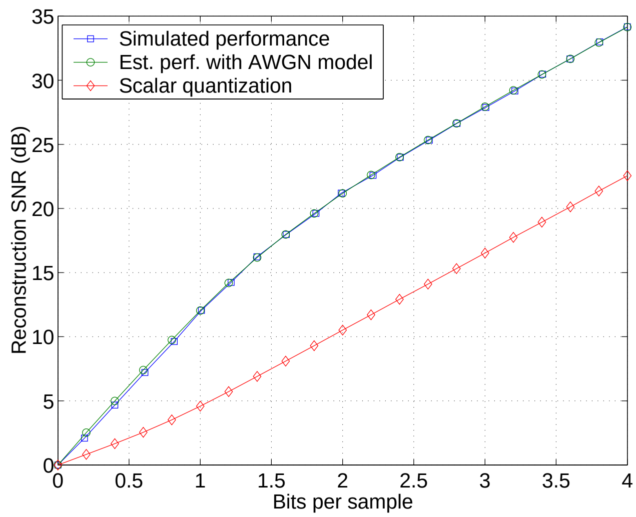

Figure 5.5. Estimation of predictive coding performance with linear AWN model. The experiment is identical to Figure 5.4, but with entropy coding of the scalar quantizer output.

Figure 5.5 shows the results for the experiment repeated with a uniform quantizer with entropy coding. The rate predicted by the AWN model plotted in the figure is based on the theoretical coding loss β for entropy coded scalar quantization of a Gaussian source. The quantizer step size is also selected assuming a Gaussian input. For the simulated output, the rate of the quantizer is based on the actual histogram of the quantizer output values in the simulation. Again, the linear AWN model predicts the true performance almost exactly.

Comparing Figures 5.4 and 5.5, entropy coding offers a small improvement in reconstruction SNR, which is mostly due to the increased coding gain for the same rate. Also, as discussed in Section 2.8.1, with entropy coding it is possible to obtain fractional rates, including rates below one bit per sample.

5.8 Relation to Rate–Distortion and Optimal Sample Rate

5.8.1 High Rate Comparison

It is interesting to compare the MSE of the predictive quantizer in Theorem 5.1 with the theoretical minimum distortion given by rate–distortion theory. To this end, consider a source $z[k]$ generated by the state space model (5.9). Since we assume that $w_z[k]$ is zero-mean, white noise with unit variance, $z[k]$ has a power-spectral density given by

$$S_z(z) = |H_z(z)|^2,$$

where $H_z(z)$ is the transfer function of the mapping from $w_z[k]$ to $z[k]$,

$$H_z(z) = C(zI - A)^{-1}B + D.$$

Now, suppose one wishes to quantize $z[k]$ with an average rate of R bits per sample. A classical result of rate–distortion theory (see, *e.g.*, [11]) is that, when R is sufficiently high, the minimum average distortion per sample is given by

$$D_{\min}(R) = 2^{-2R} F(S_z), \tag{5.23}$$

where $F(H_z)$ is defined as

$$F(S_z) = \exp\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} \log S_z(e^{i\theta})d\theta\right].$$

The formula (5.23) can be derived by dividing the source into various frequency bands, and then applying a standard water-filling argument. To compare the distortion (5.23) with the MSE of the predictive quantizer, we use the classical Szegö identity (see [20] or [77]) that

$$F(S_z) = \mathbf{Tr}(CP_0C' + DD') \tag{5.24}$$

where P_0 is the solution to the algebraic Riccati equation,

$$\begin{aligned} P_0 &= AP_0A' - (AP_0C' + BD')G_0(CP_0A' + DB') + BB' \\ G_0 &= (CP_0C' + DD')^{-1}. \end{aligned} \tag{5.25}$$

The matrix P_0 in the Szegö identity has the interpretation of the one-step ahead prediction error variance of the source $z[k]$. Substituting (5.24) into (5.23), we see that the theoretical minimum distortion at a high rate R is given by

$$D_{\min}(R) = 2^{-2R}\mathbf{Tr}(CP_0C' + DD'). \tag{5.26}$$

In comparison, the distortion for the optimal predictive quantizer in Theorem 5.1 is

$$D_{\text{pred}} = \beta\mathbf{Tr}(CPC' + DD'), \tag{5.27}$$

where P is the solution to the modified algebraic Riccati equation (5.14). Comparing (5.27) with (5.26), we see two differences between the MSE of the predictive quantizer and the minimum theoretical MSE:

- The factor 2^{-2R} in the expression (5.26) for the theoretical minimum MSE, is replaced by the scalar quantizer coding loss β in the expression (5.27) for the predictive quantizer. As discussed in Section 2.8.1, the parameter β depends on the specific scalar quantizer design. Optimal uniform scalar quantization followed with entropy coding can achieve an asymptotic coding loss of $\beta = (\pi e/6)2^{-2R}$. Thus, using such a quantizer, the MSE of the predictive quantizer would differ from the theoretical minimum by a constant factor $\pi e/6$, or approximately 1.54 dB.

  Moreover, since optimal vector quantization achieves a coding loss β exactly equal to 2^{-2R}, there would be no gap in the initial factors of both expressions, if it were possible to replace the scalar quantizer $Q(\cdot)$ in the predictive quantizer with an optimal vector quantizer. Using vector quantization with predictive coding is difficult in practice: General vector quantization has exponential complexity in the number of samples that are coded together. Moreover, to maintain causality without restricting the vector quantizer structure would require buffering before the quantizer and a less effective prediction filter.

- The second difference between expressions (5.26) and (5.27) is in the terms P_0 and P. Comparing the algebraic Riccati equation (5.25) and modified algebraic equation (5.14), we see that P is equal to P_0 precisely when $\beta = 0$. In general, one can show that P is equal to P_0 up to first order: that is,

$$P = P_0 + O(\beta).$$

  Substituting this expression into (5.27), we obtain

$$D_{\text{pred}} = \beta \mathbf{Tr}(CP_0C' + DD') + O(\beta^2),$$

  so the difference between P_0 and P results in a second-order effect that vanishes at small values of β (i.e. large number of bits per sample, R).

  The difference between P_0 and P can also be understood as follows: As mentioned above, P is the one-step ahead prediction variance of estimating the state $x[k]$ from the past samples $z[j]$ for $j < k$. Similarly, P_0 is the one-step ahead prediction error for $x[k]$ from the past quantized samples $q[j]$ for $j < k$. The error variances P and P_0 are sometimes referred to as the *open-loop* and *closed-loop* prediction variances, to reflect that the latter is performed in closed-loop with the quantizer.

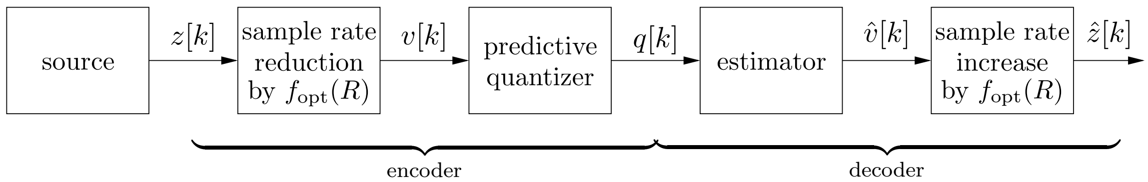

Figure 5.6. Predictive quantization with downsampling.

We conclude that, at high rates R, predictive quantization as discussed here is optimal except for two factors: a term due to the imperfection of scalar quantization, and, as a small second-order effect, a term due to the quantization noise in the feedback loop.

5.8.2 General Rate Analysis

The previous analysis was for high rate R. We now consider the case for a general rate. For simplicity, suppose that the power spectral density, $S(e^{i\theta})$, is monotonically decreasing in $|\theta|$ in the interval $\theta \in [-\pi, \pi]$. This condition would occur if the signal $z[k]$ is lowpass with the signal's power content decreasing with frequency. With this assumption, the minimum average distortion per sample for a general rate R is given by

$$D_{\min}(R) = 2^{-2R/f_{\text{opt}}(R)} F(S_z, f_{\text{opt}}(R)) + G(S_z, f_{\text{opt}}(R)), \tag{5.28}$$

where $f_{\text{opt}}(R)$ is given by,

$$f_{\text{opt}}(R) = \max\left\{ f \in [0,1] \mid \frac{1}{2\pi} \int_{|\theta|<\pi f} \log_2\left(S_z(e^{i\theta})/S_z(e^{if})\right) d\theta \le 2R \right\} \tag{5.29}$$

and

$$\begin{aligned} F(S_z, f) &= \exp\left[\frac{1}{2\pi f} \int_{|\theta|<\pi f} \log S_z(e^{i\theta}) d\theta \right], \\ G(S_z, f) &= \frac{1}{2\pi} \int_{|\theta|>\pi f} S_z(e^{i\theta}) d\theta, \end{aligned}$$

and the integrals are defined over the interval $\theta \in (-\pi, \pi)$. The minimization in (5.28) again arises from a water filling argument. The set of digital frequencies, θ with $|\theta| < \pi f_{\text{opt}}(R)$ represent the digital frequencies where it is optimal to use non-zero rate, and the term $e^{-2R/f_{\text{opt}}(R)} F(S_z, f_{\text{opt}}(R))$ is the resulting distortion on these frequencies. The term $G(S_z, f_{\text{opt}}(R))$ represents the residual distortion on the frequencies $|\theta| > \pi f_{\text{opt}}(R)$.

For a general rate R, the distortion (5.27) from a direct application of predictive quantization may be significantly larger than the minimum distortion (5.28). However, consider quantization of the signal $z[k]$ depicted in Figure 5.6. The quantization is performed in three steps:

(a) The signal $z[k]$ is filtered at the cutoff frequency $f_{\text{opt}}(R)$, and then downsampled by a factor $f_{\text{opt}}(R)$ to yield $v[k]$.

(b) Predictive quantization is then applied to the signal $v[k]$ with an average rate of $R/f_{\text{opt}}(R)$ bits per sample yielding the quantized data $q[k]$. The quantizer output, $q[k]$, is at a rate of $1/f_{\text{opt}}(R)$ times the rate of the samples of the original samples $z[k]$. Thus, the total bit rate of the data $q[k]$ is still R bits per sample or $z[k]$.

(c) The decoder receives the samples $q[k]$ reconstructs an estimate $\hat{v}[k]$ of the downsampled signal $\hat{v}[k]$.

(d) Finally, $\hat{v}[k]$ is upsampled by a factor $1/f_{\text{opt}}(R)$ to yield the final quantized signal $\hat{z}[k]$.

With ideal downsampling, it can be shown that the resulting quantization error from this four step process will be given by,

$$D_{\text{pred}}(R) = \beta(R/f_{\text{opt}}(R))F_{cl}(S_z, f_{\text{opt}}(R)) + G(S_z, f_{\text{opt}}(R)). \tag{5.30}$$

The second term, $G(H_z, f_{\text{opt}}(R))$, is the signal energy lost in the lowpass filtering in step (a). The first term is the quantization error of the downsampled signal and itself has two factors: the quantization coding loss, $\beta(R/f_{\text{opt}}(R))$, and the closed-loop prediction error, $F_{cl}(H_z, f_{\text{opt}}(R))$. The quantization coding loss is shown as a function of the bits per sample $R/f_{\text{opt}}(R)$ at the downsampled rate. Comparing (5.30) with (5.28) we see that the error variance with predictive quantization differs from the minimum achievable distortion $D_{\min}(R)$ by two factors: (a) the scalar quantization coding loss $\beta(R/f_{\text{opt}})$ is replaced by the optimal vector quantization coding loss $2^{-2R/f_{\text{opt}}}$; and (b) the closed-loop prediction error variance $F_{cl}(S_z, f_{\text{opt}})$ is replaced by the open-loop prediction error $F(S_z, f_{\text{opt}})$. The two differences are identical to those in the high rate case. We conclude that by appropriately downsampling the signal, predictive quantization will again be optimal except for the loss in scalar quantization and, as a smaller effect, the additional error for closed-loop prediction.

A simple intuition for this result is as follows: Recall that a prediction filter whitens the spectrum of the quantizer input. At frequencies at which the power spectral density is so low that the bit allocation is zero, it is better to discard the signal by sampling rate conversion than to amplify for whitening.

The fact that predictive quantization fares poorly compared to the rate–distortion bound at low rates (*i.e.*, has a performance gap larger than the $\frac{1}{2}\log_2(\pi e/6) \approx 0.255$ bits that appears at high rates) had been observed previously by Guleryuz and Orchard [78]. They improve the low-rate performance of DPCM by pre- and post-filtering, but without sampling rate conversion. Sampling rate reduction seems to be an easier and more effective way to improve upon conventional DPCM; this merits further study.

Chapter 6

Robust Predictive Quantization

This chapter uses the Markov jump linear system estimation framework of Chapter 3 and the state space representation of predictive quantization of Chapter 5 to develop a theory for robust predictive quantization. We optimize the encoder filter for overall performance (including the effect of losses along with quantization) rather than for minimum prediction error variance. Importantly, the analysis is for a system in which the decoder knows the channel state; this is both more challenging and, in part because of the ubiquity of packetized communication, more relevant than a system with a time-invariant decoder.

In making predictive quantization robust to losses of transmitted data, the prevailing methods are focused entirely on the decoder. When the source is a stationary Gaussian process, the optimal estimator is clearly a Kalman filter. Authors such as Chen and Chen [23] and Gündüzhan and Momtahan [79] have extended this to speech coding and have obtained significant improvements over simpler interpolation techniques.

One line of related research uses residual source redundancy to aid in channel decoding [6, 17, 24, 80, 131]. In stark contrast to the work reported in this dissertation, the prior works are all focused on discrete-valued sources sent over discrete memoryless channels. Most of them assume ARQ of incorrectly decoded blocks. Here we consider the entire end-to-end communication problem, with a continuous-valued source and potentially complicated source and channel dynamics.

Buch *et al.* [17] go so far as to provocatively suggest a complete lack of compression as an alternative to channel coding. It is plain that there is some middle ground: compressing somewhat (i.e., not as completely as you could) still leaves some redundancy to exploit at

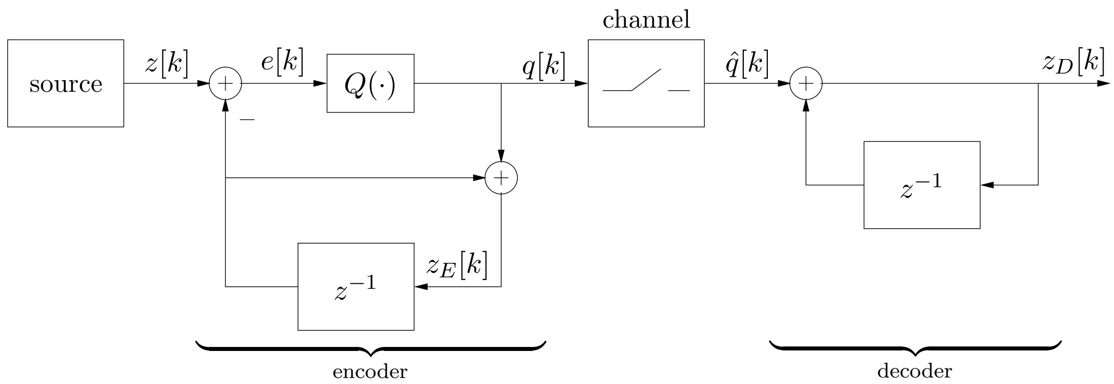

Figure 6.1. First-order predictive quantizer encoder and decoder with a lossy channel.

the decoder to mitigate losses. However, until now there has apparently been no design methodology for optimizing the encoder filter for closed-loop operation with possible losses. Not only that, there was no efficient computation for the MSE performance given an encoder filter. Here we optimize the encoder filter for MSE accounting for both source and channel dynamics, in a computationally efficient manner, rather than minimizing prediction error variance.

6.1 Prediction vs. Robustness: First-Order Example

One major problem with predictive quantization is its lack of robustness to losses of the quantizer samples. To see this potential problem, consider again the simple first-order predictive quantizer in Figure 5.1, but suppose that the samples $q[k]$ must be communicated over a lossy channel. The lossy system is illustrated in Figure 6.1. As before, the first-order predictive quantizer encoder can be described by the recursive equations

$$\begin{aligned} q[k] &= Q(z[k] - z_E[k-1]) \\ z_E[k] &= z_E[k-1] + q[k]. \end{aligned} \tag{6.1}$$

Here, we have added the subscript to z_E to denote the encoder signal.

Now, if the channel were lossless, all the samples $q[k]$ would be known to the decoder, and the decoder could therefore exactly reconstruct $z_E[k]$. However, with losses, the decoder must merely estimate $z_E[k]$. As a simple estimator, let us say the decoder is implemented as an integrator similar to the encoder,

$$z_D[k+1] = z_D[k] + \hat{q}[k] \tag{6.2}$$

where $\hat{q}[k]$ is an estimate of the sample $q[k]$,

$$\hat{q}[k] = \begin{cases} q[k] & \text{when } q[k] \text{ is not lost} \\ 0 & \text{when } q[k] \text{ is lost.} \end{cases}$$

If there are no losses, $\hat{q}[k] = q[k]$ and $z_D[k] = z_E[k]$ for all k. However, if there are any losses, the decoder signal $z_D[k]$ and encoder signal $z_E[k]$ will begin to diverge. In fact, if we subtract (6.1) and (6.2), we obtain the recursion

$$z_E[k+1] - z_D[k+1] = z_E[k] - z_D[k] + q[k] - \hat{q}[k].$$

Therefore, errors due to losses accumulate in the difference between the encoder and decoder signals. Indeed, if there is even a single loss, the error will persist for all time. Clearly, this simple first-order encoder-decoder system lacks any robustness.

One possible way to rectify the problem is with a modified encoder of the form

$$\begin{aligned} q[k] &= Q(z[k] - \alpha z_E[k-1]), \\ z_E[k] &= \alpha z_E[k-1] + q[k], \end{aligned} \tag{6.3}$$

for some $\alpha \in [0, 1)$. The decoder can use a similar recursion,

$$z_D[k+1] = \alpha z_D[k] + \hat{q}[k]. \tag{6.4}$$

In this system,

$$z_E[k+1] - z_D[k+1] = \alpha(z_E[k] - z_D[k]) + q[k] - \hat{q}[k],$$

so losses decay exponentially with a factor α. Thus, selecting α closer to zero makes the effect of losses decay faster, thereby improving the robustness of the system. In particular, losses do not persist indefinitely. Using such an encoder and decoder with $\alpha < 1$ is an example of what we call a *robust predictive quantizer*.

There is a price for the robustness. The factor α determines how much of the previous sample is used in the encoder, and thus represents how much prediction will be exploited. If $\alpha = 1$, the encoder attempts to maximally utilize the information from the previous sample and thereby maximize the prediction gain. But, as we have seen, losses accumulate and persist indefinitely. At the other extreme, if $\alpha = 0$, there is no persistence of the effect of losses. But, the encoder uses no prediction and reduces to a simple scalar quantizer.

We therefore see that there is a fundamental trade-off between prediction gain and robustness. In this example, depending on the loss probabilities and source statistics, there is some optimal α between 0 and 1.

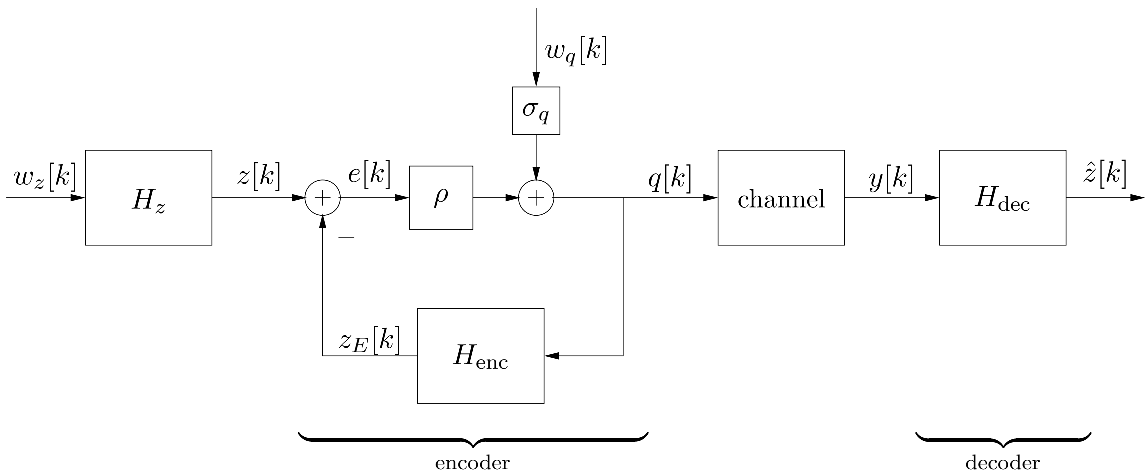

Figure 6.2. General predictive quantizer encoder and decoder with a lossy channel. The quantizer is modeled as a linear gain with AWGN noise.

This chapter explores this optimization between prediction gain and robustness for general predictive quantizers. We will attempt to compute an optimal robust prediction gain that is analogous to the factor α, but applies to higher order coders. Our analysis will combine the state space modeling of Chapter 5 along with the LMI analysis in Chapters 3 and 4. Our analysis and optimization are original, and there are no analogous results in frequency-domain analysis. One might pursue a frequency-domain analysis for a case in which the decoder uses an LTI filter; however, the resulting performance would be considerably worse than in our system, where the decoder is time varying (with dependence on $\theta[k]$). The jump linear state space framework is the best tool for the problem of practical relevance.

6.2 State space Model

The general problem we consider is shown in Figure 6.2. As in Chapter 5, the source to be modeled is denoted $z[k]$, and assumed to be the output of a stable, LTI system, H_z, with a state space realization,

$$\begin{aligned} x_z[k+1] &= Ax_z[k] + Bw_z[k] \\ z[k] &= Cx_z[k] + Dw_z[k], \end{aligned} \tag{6.5}$$

where $w_z[k]$ is zero-mean white noise with unit variance. Also, as before, the encoder is realized as a scalar quantizer in feedback with an LTI prediction filter. We will assume that the prediction filter has the form,

$$\begin{aligned} x_E[k+1] &= Ax_E[k] + Lq[k] \\ z_E[k] &= Cx_E[k], \end{aligned} \tag{6.6}$$

where $q[k]$ is the input from the quantizer and $z_E[k]$ is the prediction filter output. The matrix L is the gain of the prediction filter. If L is designed as in Theorem 5.1, H_{pred} will obtain the maximum prediction gain assuming there are no losses of the samples $q[k]$ between the encoder and decoder. In this chapter, where we consider losses, we will treat L as a design parameter that is optimized to trade off the prediction gain and robustness.

Also, as in Chapter 5, we will model the quantizer as a linear gain with additive noise,

$$q[k] = \rho(z[k] - z_E[k]) + \sigma_q w_q[k] \tag{6.7}$$

where $w_q[k]$ is assumed to be zero-mean white noise with unit variance. Following Lemma 5.1, we assume the gain ρ and quantization noise variance σ_q^2 satisfy

$$\begin{aligned} \rho &= 1 - \beta \\ \sigma_q^2 &= \beta(1-\beta)\mathbf{E}|z[k] - z_E[k]|^2, \end{aligned} \tag{6.8}$$

for a given quantization coding loss $\beta \in [0, 1]$.

For the channel, we will use the Markov erasure model discussed earlier in Section 4.2. Specifically, we will assume that the channel erasures are governed by a stationary Markov chain $\theta[k] \in \{0, \ldots, M-1\}$ independent of $w_z[k]$ and $w_q[k]$. The encoder output sample $q[k]$ will be erased when $\theta[k] \in I_{loss}$ where I_{loss} is some subset of the Markov states $\{0, \ldots, M-1\}$. We will denote the channel output by $\hat{q}[k]$ and take the output to be zero when the sample is lost. Thus,

$$\hat{q}[k] = \begin{cases} q[k] & \text{when } \theta[k] \notin I_{loss} \\ 0 & \text{when } \theta[k] \in I_{loss}. \end{cases} \tag{6.9}$$

As before, we will denote by p_{ij} the Markov transition probabilities,

$$p_{ij} = \mathbf{Pr}\left(\theta[k+1] = j \mid \theta[k] = i\right). \tag{6.10}$$

We will assume that the Markov chain is aperiodic and irreducible so that there is a unique stationary distribution,

$$q_i = \mathbf{Pr}\left(\theta[k] = i\right), \tag{6.11}$$

satisfying the equality,

$$q_j = \sum_{i=0}^{M-1} p_{ij} q_i, \quad j = 0, \ldots, M-1.$$

6.3 Jump Linear Decoding and Analysis

6.3.1 Encoder Gain Matrix Feasibility

Before discussing the decoder structure, it is useful to discuss our assumptions on the encoder in more detail. In the previous section, we fixed the prediction filter in the encoder to be of the form (6.6). This prediction filter has the same form as the optimal filter for the lossless system, except that we can use any gain matrix L. In this way, we treat L as a design parameter that can be optimized depending on the loss and source statistics.

In fixing the encoder to be of the form (6.6), we have eliminated certain degrees of freedom in the design. Specifically, the LMI framework would allow the matrices A and C in (6.6) to differ from the corresponding quantities in (6.5). But since the optimization philosophy in Section 6.4 would lead us to (6.6), we impose this form at the onset to simplify the following discussion.

The following definition specifies certain basic properties of a gain matrix L to be considered for the encoder.

Definition 6.1 *Consider a predictive quantizer system in Figure 6.2. Suppose the source signal $z[k]$ is generated by a stable LTI system of the form (6.5), the prediction filter can be described by (6.6), and the quantizer can be described by (6.7). Let $\beta \in [0,1]$ be the quantization coding loss, so that $\rho = 1-\beta$. For this system, an encoder gain matrix L will be called* feasible *if*

(a) the resulting closed-loop map from $(w_z, w_q) \mapsto (x_z, x_E)$ is well-posed and stable; and

(b) there exists a quantization noise level σ_q such that when $w_q[k]$ and $w_z[k]$ are zero-mean, white noise with unit variance, the closed-loop signal $z[k] - z_E[k]$ satisfies

$$\sigma_q^2 = \beta(1-\beta) \lim_{k\to\infty} \mathbf{E}\|z[k] - z_E[k]\|^2.$$

The definition is similar to Definition 5.1 from our discussion of predictive encoders. As before, we treat the quantization noise σ_q not as a given, but rather as a function of the encoder gain L and quantization coding loss β. The noise level must be determined from the closed-loop system. The following proposition provides a characterization of the set of feasible encoder gains, and their corresponding quantization noise levels.

Proposition 6.1 *Consider the predictive quantizer system in Definition 6.1. Given an encoder gain matrix L, suppose there exists a $Q \geq 0$ satisfying the Lyapunov equation,*

$$Q = (A - \rho LC)' Q (A - \rho LC) + C'C, \tag{6.12}$$

with

$$\beta(1-\beta) L' Q L < 1. \tag{6.13}$$

Then L is feasible and the corresponding quantization noise level is given by,

$$\sigma_q^2 = \frac{\beta(1-\beta)}{1 - \beta(1-\beta)L'QL} \mathbf{Tr}\left((B - \rho LD)'Q(B - \rho LD) + D'D\right). \tag{6.14}$$

Proposition 6.1 provides a simple way of testing the feasibility of a candidate gain matrix L and determining the corresponding quantization noise level. Specifically, the gain matrix L is feasible if the solution Q to the Lyapunov equation (6.12) satisfies $Q \geq 0$ and the condition in (6.13). If the gain matrix L is feasible, the closed-loop quantization noise level is given by (6.14).

In this way, Proposition 6.1 characterizes the feasible gain matrices L, and their corresponding quantization noise. Now, recall from our discussion in the previous chapter that, with no channel losses, the optimal encoder for the lossless system should minimize the resulting quantization noise variance σ_q^2. Therefore, the expression for L in Theorem 5.1 is precisely the gain matrix that minimizes σ_q^2 in Proposition 6.1.

However, in the presence of losses, we will see that the optimal gain does not necessarily minimize the quantization noise. The quantizer input variance can be seen as a measure of how much energy the prediction filter subtracts out from the previous quantizer output samples. With losses, the optimal filter may not subtract out all the energy, thereby leaving some redundancy in the quantizer output samples and thus improving the robustness to losses.

Proof of Proposition 6.1: As usual, define the error signals, $e_x[k] = x_z[k] - x_E[k]$ and $e_z[k] = z[k] - z_E[k]$. Combining (6.5), (6.6) and (6.7), we obtain

$$\begin{aligned} e_x[k+1] &= (A - \rho LC)e_x[k] + (B - \rho LD)w_z[k] - \sigma_q L w_q[k] \\ e_z[k] &= Ce_z[k] + Dw_z[k]. \end{aligned} \tag{6.15}$$

The system (6.15) is a standard LTI system.

Now, suppose there exists a $Q \geq 0$ satisfying (6.12). By the assumption of Definition 6.1, A is stable. Therefore, for any matrix C and L, $(A - \rho LC, C)$ is detectable. Consequently, by Proposition 2.4, the existence of a matrix $Q \geq 0$ satisfying (6.12) implies that $A - \rho LC$ is stable. This in turn implies that, for any σ_q, the mapping $(w_z, w_q) \mapsto e_x$ is stable. Also, the since A is stable, the map $w_z \mapsto x$ is stable. Therefore, since $e_x = x - x_E$, the mapping $(w_z, w_q) \mapsto (x, x_E)$ is well-posed and stable and the gain matrix L satisfies condition (a) of the feasibility requirements in Definition 6.1.

For condition (b), we must show that if σ_q is defined as in (6.14), the resulting closed system satisfies,

$$\sigma_q^2 = \beta(1-\beta)\sigma_z^2,$$

where

$$\sigma_z^2 = \lim_{k\to\infty} \mathbf{E}|z[k] - z_E[k]|^2 = \lim_{k\to\infty} \mathbf{E}|e_z[k]|^2.$$

From Proposition 2.3 and (6.14),

$$\begin{aligned}
&\sigma_q^2 - \beta(1-\beta)\sigma_z^2 \\
&= \sigma_q^2 - \beta(1-\beta)\mathbf{Tr}\left((B-\rho LD)'Q(B-\rho LD) + D'D + \sigma_q^2 L'QL\right). \\
&= (1-\beta(1-\beta)L'QL)\sigma_q^2 \\
&- \beta(1-\beta)\mathbf{Tr}\left((B-\rho LD)'Q(B-\rho LD) + D'D\right) = 0
\end{aligned}$$

Hence, $\sigma_q^2 = \beta(1-\beta)\sigma_z^2$ and, thus, σ_q^2 is the quantization noise level for L. □

6.3.2 Jump Linear Decoder

Having characterized the set of feasible encoders, we can now consider the decoder. As discussed in the previous chapter, the decoder can be seen as a Kalman filtering problem: the decoder must estimate the source signal $z[k]$ from the quantizer samples $q[k]$ that are not lost. The two signals are generated by a system with two states: $x_z[k]$ in the source signal model, and $x_E[k]$ in the prediction filter.

In the lossless system, all the samples $q[k]$ are known to the decoder and, consequently, the decoder can completely reconstruct the prediction filter state $x_E[k]$. Therefore, the Kalman filter in the decoder need only estimate the source signal state $x_z[k]$.

However, in the presence of losses, the prediction filter state $x_E[k]$ will not, in general, be known exactly to the decoder. Consequently, the optimal decoder must estimate both the source signal state $x_z[k]$ *and* prediction filter state $x_E[k]$. Therefore, to describe the decoder as a Kalman filter, we need to express the source signal $z[k]$ and quantizer output $q[k]$ as outputs of a single state space system with states $x_z[k]$ and $x_E[k]$.

To this end, define the joint state,

$$x[k] = [x_z[k]' \; x_E[k]']',$$

which contains the unknown states of both the source signal system and prediction filter. Also, let $w[k]$ denote the joint noise vector,

$$w[k] = [w_z[k]' \; w_q[k]']',$$

which contains both the source signal input and quantization noise.

Now, the decoder must estimate the state $x[k]$ and signal $z[k]$ from the signal $\hat{q}[k]$ in (6.9), which is the sequence of predictive quantizer samples not lost in the erasure channel. We will assume that the channel state $\theta[k]$ is known to the decoder, so the decoder knows which samples have been erased.

There are two cases for the state and output equations for the system: when the sample $q[k]$ is received by the decoder, and when the sample is lost. We will first consider the case when the sample $q[k]$ is not lost. In this case, $\hat{q}[k] = q[k]$. Using this fact along with (6.5), (6.6) and (6.7), we obtain the larger state space system,

$$\begin{aligned} x[k+1] &= A_{NL}x[k] + B_{NL}w[k] + u[k] \\ z[k] &= C_1 x[k] + D_1 w[k] \\ \hat{q}[k] &= C_2 x[k] + D_2 w[k] \end{aligned} \tag{6.16}$$

where

$$\begin{aligned} A_{NL} &= \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}, & B_{NL} &= \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix}, \\ C_1 &= \begin{bmatrix} C & 0 \end{bmatrix}, & D_1 &= \begin{bmatrix} D & 0 \end{bmatrix}, \\ C_2 &= \begin{bmatrix} \rho C & -\rho C \end{bmatrix}, & D_2 &= \begin{bmatrix} \rho D & \sigma_q \end{bmatrix}, \\ u[k] &= Uq[k], & U &= \begin{bmatrix} 0 \\ L \end{bmatrix}. \end{aligned}$$

The system (6.16) expresses the source signal $z[k]$ and the channel output $\hat{q}[k]$ as outputs of a single larger state space system. The system has two inputs: the noise vector $w[k]$ and the signal $u[k]$. When the sample $q[k]$ is not lost over the channel, the input $u[k]$ is known to the decoder. Hence we have added the subscript NL on the matrices, A_{NL} and B_{NL} to indicate the "no loss" matrices.

When the sample $q[k]$ is lost,

$$\begin{aligned} x[k+1] &= A_L x[k] + B_L w[k] \\ z[k] &= C_1 x[k] + D_1 w[k] \\ \hat{q}[k] &= 0 \end{aligned} \tag{6.17}$$

where

$$A_L = A_{NL} + UC_2, \quad B_L = B_{NL} + UD_2.$$

Thus, from the perspective of the decoder, the encoder system alternates between two possible models: (6.16) when the sample $q[k]$ is not lost in the channel, and (6.17) when the sample is lost. Now, in the model we have assumed in Section 6.2, the loss event occurs when the Markov state $\theta[k]$ enters a subset of the discrete states denoted I_{loss}. We can thus write the encoder system as a jump linear system driven by the channel Markov state $\theta[k]$: When $\theta[k] = i$,

$$\begin{aligned} x[k+1] &= A_i x[k] + B_i w[k] + u_i[k] \\ z[k] &= C_1 x[k] + D_1 w[k] \\ \hat{q}[k] &= C_{i2} x[k] + D_{i2} w[k], \end{aligned} \tag{6.18}$$

where the system matrices are given by,

$$(A_i, B_i, C_{i2}, D_{i2}) = \begin{cases} (A_{NL}, B_{NL}, C_2, D_2) & \text{when } i \notin I_{loss}, \\ (A_L, B_L, 0, 0) & \text{when } i \in I_{loss} \end{cases}$$

and $u_i[k]$ is the know input

$$u_i[k] = \begin{cases} U\hat{q}[k] & \text{when } i \notin I_{loss}, \\ 0 & \text{when } i \in I_{loss}. \end{cases}$$

The matrices C_1 and D_1 do not vary with $\theta[k]$.

Following Section 3.6, we can consider a jump linear estimator of the following form: When $\theta[k] = i$,

$$\begin{aligned} \hat{x}[k+1] &= A_i \hat{x}[k] + L_{i1}(\hat{q}[k] - C_{i2}\hat{x}[k]) + u_i[k] \\ \hat{z}[k] &= C_1 x[k] + L_{i2}(\hat{q}[k] - C_{i2}\hat{x}[k]) \end{aligned} \tag{6.19}$$

where L_{i1} and L_{i2} are gain matrices that are to be determined by the optimization.

Before considering the optimization, it is useful to compare the jump linear estimator (6.19) with the decoder for the lossless case in (5.15). The most significant difference is that the decoder in (6.19) must estimate the states, $x_z[k]$, for the source signal system, as well as the encoder states, $x_E[k]$. As discussed above, in the lossless case, the encoder states are known to the decoder and do not need to be estimated. Therefore, if the lossless problem must estimate a state of dimension n, the lossy estimator must estimate a state of dimension $2n$.

A second difference is the gain matrices L_{i1} and L_{i2}. In the decoder (5.15) for the lossless channel, the gain matrices are constant. Moreover, $L_1 = L$ where L is the optimal encoder gain matrix, and $L_2 = I$. In the jump linear decoder (6.19), the gain matrices L_{i1} and L_{i2} vary with the Markov state $\theta[k]$ and do not necessarily have any simple relation with the encoder matrix.

6.3.3 LMI analysis

Having modeled the system and decoder as jump linear systems, we now can find the optimal decoder gain matrices L_{i1} and L_{i2} in (6.19) using the LMI analysis of Chapter 3. Specifically, Theorem 3.2 in that chapter provides an LMI optimization for computing the gain matrices L_{i1} and L_{i2} that minimizes the asymptotic MSE,

$$\mathrm{MSE} = \lim_{k\to\infty} \mathbf{E}|z[k] - \hat{z}[k]|^2, \tag{6.20}$$

where the expectation is over the random signal $z[k]$, the quantization noise $w_q[k]$ and Markov state sequence $\theta[k]$. For the decoder problem, the MSE in (6.20) is precisely the asymptotic mean-squared reconstruction error between the original signal $z[k]$ and the decoder output $\hat{z}[k]$.

The LMI method thus provides a simple way of computing the asymptotic minimum reconstruction error for a given source signal model, predictive encoder and channel loss statistics. By varying the channel loss model parameters, one can thus quantify the effect of channel losses on a predictive quantization system with a given encoder.

The overall analysis algorithm can be described as follows:

Algorithm 6.1 (LMI analysis) *Consider a predictive quantizer system in Figure 6.2. Suppose the source signal $z[k]$ is generated by a stable LTI system of the form (6.5), the prediction filter can be described by (6.6) for a given gain matrix L, and the quantizer can be described by (6.7). Let $\beta \in [0,1]$ be the quantization coding loss, so that $\rho = 1 - \beta$.*

Suppose the channel erasures can be described by an M-state Markov chain $\theta[k]$ satisfying Assumption 3.1. Then, the optimal decoder of the form (6.19) can be computed as follows:

1. *Use Proposition 6.1 to determine if the encoder gain matrix L is feasible. Specifically, verify that there is a solution $Q \geq 0$ to the Lyapunov equation (6.12), and confirm that the solution satisfies (6.13).*

2. *If the gain matrix L is feasible, compute the closed-loop quantization quantization noise, σ_q^2 in (6.14).*

3. *Compute the jump linear system matrices A_i, B_i, C_{i1}, C_{i2}, D_{i1} and D_{i2} as in Section 6.3.2.*

4. *Use Theorem 3.2 to compute the optimal gain matrices L_{i1} and L_{i2} for the decoder (6.19) and the corresponding asymptotic MSE,*

$$MSE = \lim_{k\to\infty} \mathbf{E}|z[k] - \hat{z}[k]|^2.$$

6.4 Encoder Gain Optimization

Algorithm 6.1 in the previous section describes how to compute the minimum MSE achievable at the decoder for a *given* predictive encoder. However, to maximize the robustness of the overall quantizer system, one would like to select the predictive encoder that minimizes this MSE.

To be more specific, let MSE(L) be the minimum achievable MSE for a given encoder gain matrix L. This minimum error, MSE(L), can be computed from Algorithm 6.1 given a model for the signal and a Markov loss model for the channel. Ideally, one would like to search over all gain matrices L to minimize MSE(L). That is, we wish to compute the optimal encoder gain matrix,

$$L_{opt} = \arg\min_{L} \mathrm{MSE}(L). \tag{6.21}$$

Unfortunately, this minimization is difficult. The function MSE(L) is, in general, a complex nonlinear function of the coefficients of L. The global minima cannot be found without an exhaustive search. If the filter (6.5) for the signal $z[k]$ has order n, then L will have n

coefficients to optimize over. Therefore, even at small filter orders, n, an exhaustive search over all possible gain matrices L will be prohibitively difficult.

To overcome this difficulty, we propose the following simple, but suboptimal, search. For each $\lambda \in [0, 1]$, we compute a candidate gain matrix $L(\lambda)$ given by,

$$L(\lambda) = E_1(\lambda)G(\lambda), \tag{6.22}$$

where $E_1(\lambda)$ and $G(\lambda)$ are derived from the solutions to the modified algebraic Riccati equations,

$$\begin{aligned} P(\lambda) &= AP(\lambda)A' - \lambda(1-\beta)E_1GE_1' + BB' \\ E_1(\lambda) &= AP(\lambda)C' + BD' \\ G(\lambda) &= (CP(\lambda)C' + DD')^{-1}. \end{aligned} \tag{6.23}$$

We can then minimize the MSE, searching over the single parameter λ,

$$L_{subopt} = \arg\min_{\lambda \in [0,1]} \mathrm{MSE}(L(\lambda)). \tag{6.24}$$

Since this minimization involves a search over a single parameter, λ, the minimization can be performed easily by simply testing a large number of candidate values in the interval $\lambda \in [0, 1]$.

Of course, the minimization (6.24) may yield encoder gain matrices strictly worse than the exhaustive minimization in (6.21). However, the suboptimal search over the candidate gain matrices $L(\lambda)$ can be motivated as follows: From Theorem 5.1, we see that when $\lambda = 1$, $L(\lambda)$ is precisely the optimal encoder gain matrix for the lossless system. For other values of λ, $L(\lambda)$, is the optimal gain matrix for a lossless channel, but with a higher effective coding loss given by,

$$\beta_{eff} = 1 - \lambda(1 - \beta).$$

As λ decreases from 1 to 0, this effective coding loss, β_{eff}, increases from β to 1. The increase in the coding loss β_{eff} represents an increase in the effective quantization noise. The logic in the suboptimal search (6.24) is that the modifications for the encoder for higher quantization noise should be qualitatively similar to the modifications necessary for channel losses. As the quantization noise increases, the prediction filter is less able to rely on past quantized samples and will naturally decrease the weighting of those samples in the prediction output. This removal of past samples from the prediction output will leave redundancy in the quantized samples and should improve the robustness to channel losses.

We have seen this connection of quantization error and channel losses before: In the remarks after Theorem 5.1, we showed that the optimal prediction filter for a predictive

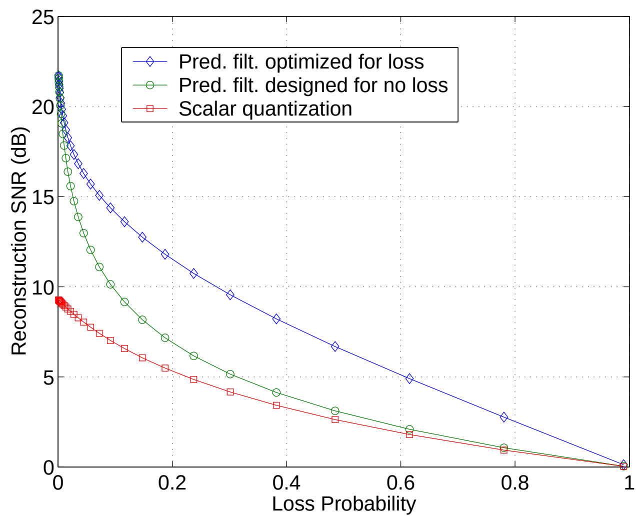

Figure 6.3. Quantization and reconstruction of a lowpass signal with i.i.d. channel losses. Plotted is the reconstruction SNR for three quantizers: (a) scalar quantization with no interpolation at the decoder for lost samples; (b) the optimal predictive encoder and decoder assuming no channel losses; and (c) the proposed robust predictive quantizer where the encoder and decoder are optimized for each loss probability. In all cases, the signal is a lowpass second-order ARMA Gaussian random process whose bandwidth is 0.1 times the sample rate. All encoders have a rate of 2 bits per sample and use uniform scalar quantization.

quantization system with no channel losses and quantization coding loss of β is precisely the same as the optimal decoder from data with no quantization error but with i.i.d. channel losses with loss probability β. Similarly, the idea in the optimization (6.24) is that the encoder designed for extra quantization error should be well-suited for channel losses.

6.5 Numerical Example

As an illustration of the robust predictive quantization design method, we return to the example of Section 5.7. As in that example, we consider the quantization of a lowpass signal $z[k]$ modeled as the output of a second-order Chebyshev filter whose bandwidth is 0.1 times the sample rate. For the scalar quantizer, $Q(\cdot)$, we assume a uniform quantizer with 2 bits per sample.

Figure 6.3 shows the reconstruction SNR for various encoders. For each encoder, the reconstruction SNR is plotted as a function of the channel loss probability assuming the channel losses are i.i.d. The reconstruction SNR is defined as the relative error

$$\text{SNR} = 10\log_{10}\left(\frac{\mathbf{E}|z[k]|^2}{\mathbf{E}|z[k]-\hat{z}[k]|^2}\right),$$

where $z[k]$ is the lowpass signal to be quantized and $\hat{z}[k]$ is the final estimate at the decoder.

The bottom curve in Figure 6.3 is the reconstruction SNR with simple scalar quantization at the encoder and no interpolation for lost samples at the decoder. That is, the decoder simply substitutes a value of zero for any lost samples. When there are no losses scalar quantization achieves a reconstruction SNR of 9.3 dB, which is the coding gain for an optimal 2 bit scalar quantizer. Since there is no interpolation at the decoder, with losses, the reconstruction error decays linearly in the loss probability.

The second curve in Figure 6.3 is the optimal encoder designed for no losses as described in Chapter 5. With no losses, this encoder achieves a reconstruction SNR of approximately 21.7 dB which represents a prediction gain of approximately 21.7 - 9.3 = 12.4 dB relative to scalar quantization. However, in the presence of losses this prediction gain rapidly decreases. For example, at a loss probability of 20%, the predictive quantizer performs less than 2 dB better than scalar quantization. In this sense, the predictive quantizer which is not designed for losses is not robust.

The top curve depicts the performance of the robust predictive quantizer where the encoder is optimized for the loss probability. Specifically, for each loss probability, the minimization (6.24) is performed to find the optimal encoder gain matrix, and the MSE at the decoder is based on the optimal jump linear decoder a computed from Algorithm 6.1.

When there are no losses, the robust predictive quantizer is identical to the optimal encoder with no losses. However, as the loss probability increases, the degradation of the robust predictive quantizer with losses is much smaller than the degradation of the encoder optimized for no loss. For example, at a 20% loss probability, the robust predictive quantizer has a reconstruction SNR more than 4.5 dB greater than the optimal lossless encoder.

The design method can also be performed for correlated losses. As an example, Figure 6.4 repeats the experiment with a Gilbert-Elliot loss model as described in Section 4.2. The good-to-bad state transition parameter, λ_1, is varied from 0 to 1, and the bad-to-good parameter set to $\lambda_2 = (1-\lambda_1)/2$. The Gilbert-Elliot model provides a simple example of a correlated error sequence. As can be seen in Figure 6.4, the robust predictive quantizer again shows an improved performance over the encoder optimized for no losses.

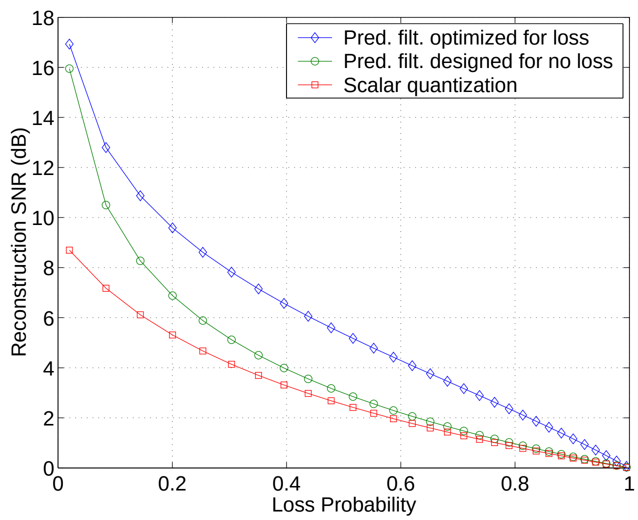

Figure 6.4. Quantization and reconstruction of a lowpass signal with Gilbert-Elliot channel losses. The signal and encoder methods are identical to Figure 6.3

Chapter 7

Multiple Description Coding by Sample Separation

Multiple description (MD) coding is a form of robust source coding where the encoded data is transmitted in multiple streams called *descriptions*. The encoding is designed so that the reconstruction of the source gracefully degrades with losses of the descriptions. MD coding has been proposed for robust communication in lossy packet networks.

The initial conception of MD coding (though not the phrase) and the first few techniques for MD coding came from Bell Labs in the late 1970s [69]. One of these techniques, proposed initially for sequences of speech samples, is a simple separation of odd-numbered samples from even-numbered samples to produce two descriptions [86, 87]. We will demonstrate the potential of our design methodology by optimizing the encoder for this MD technique. The most closely related previous work in this area [83] is severely limited in that performance optimizations are difficult and hence only attempted for a first-order Gauss-Markov source.

The explicit derivations and experimental results in this chapter are for systems with two descriptions. Thus for simplicity, the entire discussion of MD coding is given for two rather than an arbitrary number of descriptions. In fact, all but a small fraction of the literature on MD coding is for two descriptions. The generality of the developments in Chapters 4 and 6 should make it clear that larger numbers of descriptions can be handled easily. It should be noted, however, that the best techniques for more than two descriptions use constructions in which descriptions are used as side information in the decoding of other descriptions [122, 123].

This chapter is organized as follows: MD coding and its basic terminology are introduced in Section 7.1. The technique of creating two descriptions by separating odd- and even-numbered samples is described in Section 7.2. Then, Section 7.3 shows how to extend the analysis of Chapter 6 to the case where the filtered and quantized samples are lost in fixed groupings; this yields our desired design methodology. Section 7.4 summarizes our experimental results.

7.1 Introduction to Multiple Description Coding

Multiple description coding is a generalization of (ordinary) source coding with a fidelity criterion [136]. As the name suggests, the difference is to have multiple encodings of the source instead of just one. In single description coding, a source vector is encoded into a codeword (description) of expected length R bits per symbol, and reconstructing an estimate of the source from the codeword gives expected distortion per symbol D.[1] MD coding produces descriptions with rates R_1 and R_2. Having more than one description introduces the distinction between estimating the source from an individual description or from the descriptions together. This gives three possible behaviors, as shown in Figure 7.1: estimate $\{\hat{X}_k^{(1)}\}$ is computed from Description 1 alone, $\{\hat{X}_k^{(2)}\}$ from Description 2 alone, and $\{\hat{X}_k^{(0)}\}$ from both descriptions together. The expected distortion of the estimate $\{\hat{X}_k^{(i)}\}$ is denoted by D_i. Thus we find that the performance of an MD code is described by a quintuple $(R_1, R_2, D_0, D_1, D_2)$ in contrast to the pair (R, D) that describes the performance of a single-description code. The standard block diagram in Figure 7.1 illustrates why D_0 is called the *central distortion* and D_1 and D_2 are called *side distortions*.

With ordinary source coding, losing any part of the codeword can be arbitrarily bad—essentially as bad as losing the entire codeword. The point of MD coding is to provide robustness to the loss of encoded data, at least at the granularity of descriptions; that is, to make D_1 and D_2 relatively small. The trade-off in MD coding is in making the descriptions good individually versus making them good collectively.

How small can D_1 and D_2 be? By rate–distortion theory, they are obviously limited by

[1] The technical meaning of attaining rate R and distortion D can be found, *e.g.*, in [41]. Since we are not studying the MD rate–distortion region, no additional formality is necessary. The interested reader is referred to [49, 154].

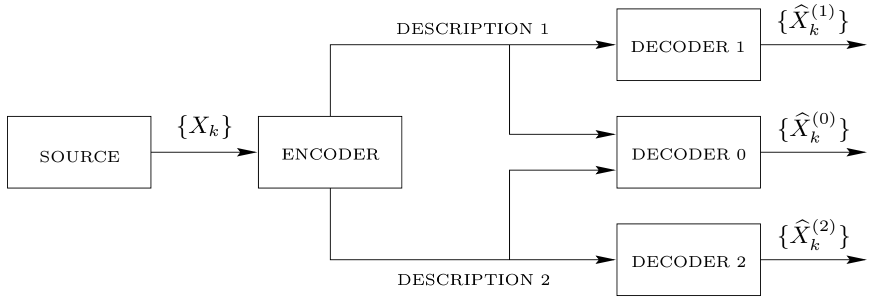

Figure 7.1. Block diagram depiction of multiple description source coding with two descriptions and three decoders.

functions of R_1 and R_2, respectively:

$$\begin{aligned} D_1 &\geq D(R_1) \\ D_2 &\geq D(R_2) \end{aligned}$$

where $D(\,\cdot\,)$ is the distortion–rate function of the source.[2] The interesting aspect of MD coding—and the inherent tension in the design of MD codes—is that D_1 and D_2 are further limited depending on how small D_0 is relative to $D(R_1 + R_2)$. Heuristically: If D_0 is close to $D(R_1 + R_2)$, then the two descriptions collectively are a very efficient representation; hence, the descriptions must be nearly independent; and from this it follows that at least one of the descriptions is not very good alone.

It is difficult to make general statements about the trade-offs in MD coding because only for an i.i.d. Gaussian source and the squared error distortion measure is the set of achievable $(R_1, R_2, D_0, D_1, D_2)$-tuples precisely known [49, 120]. For this case, the intuition above is bolstered by the bound

$$D_0 \geq 2^{-2(R_1+R_2)} \cdot \gamma_D(R_1,\, R_2,\, D_1,\, D_2),$$

where

$$\gamma_D = \frac{1}{1 - \left(\sqrt{(1-D_1)(1-D_2)} - \sqrt{D_1 D_2 - 2^{-2(R_1+R_2)}}\right)^2}$$

and the source has unit variance. The factor γ_D is at least one and it is close to one only when $D_1 D_2 - 2^{-2(R_1+R_2)}$ is not too small.

A tutorial by Goyal [69] provides an insightful illustration of the potential of MD coding for the quadratic Gaussian case, but a flaw in the figure weakens the point [70]. Follow-

[2] We are implicitly assuming that each of the $\{\hat{X}_k^{(i)}\}$'s is assessed with the same distortion measure, as is usually the case. We will further consider only mean-squared error distortion.

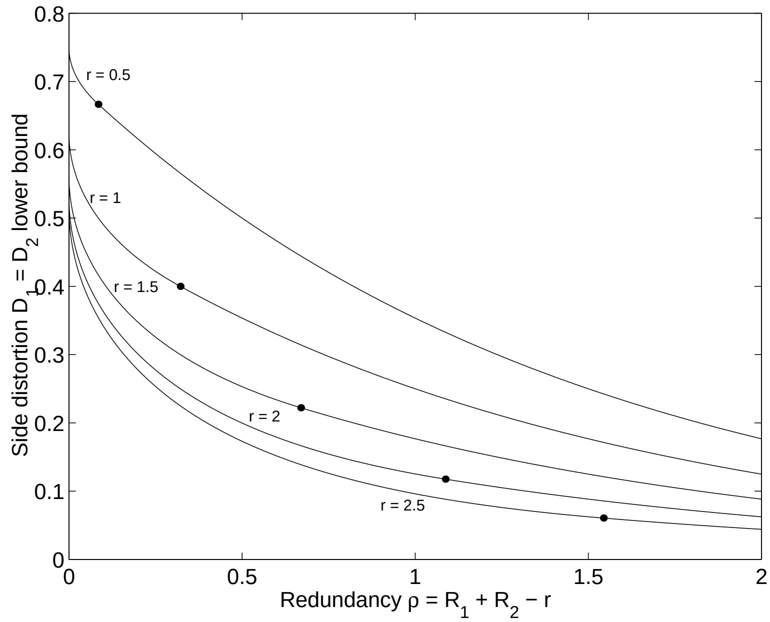

Figure 7.2. In balanced MD coding (where $R_1 = R_2$ and $D_1 = D_2$), the best possible side distortion D_1 depends on the rate in excess of the minimum needed to achieve central distortion D_0. The decay of the side distortion with *redundancy* $\rho = R_1 + R_2 - R(D_0)$ is faster than exponential for low redundancies and exponential for high redundancies. (Dots marked on the graph indicate the transition from low- to high-redundancy behavior. The figure is reproduced from [70].)

ing [69, 70], define a base rate $r = R(D_0)$ and redundancy $\rho = R_1 + R_2 - R(D_0)$. (Redundancy is rate added above what is needed to achieve D_0 with the aim of reducing D_1 and D_2.) For the balanced case, where $R_1 = R_2$ and $D_1 = D_2$,

$$D_1 \geq \begin{cases} \frac{1}{2}\left[1 + 2^{-2r} - (1 - 2^{-2r})\sqrt{1 - 2^{-2\rho}}\right], & \rho \leq \rho_T; \\ 2^{-(r+\rho)}, & \rho > \rho_T, \end{cases} \tag{7.1}$$

where $\rho_T = r - 1 + \log_2(1 + 2^{-2r})$. This bound is plotted for several values of r in Figure 7.2. To emphasize the transition between the two regimes of the bound, the ρ_Ts are marked. Figure 7.2 and expression (7.1) show an exponential decay of the lowest possible D_1 as a function of ρ, when ρ is large. The more interesting aspect is that for small ρ, the decay is faster than exponential; in fact, at $\rho = 0^+$ the slope is infinite.

As long as we restrict our attention to squared error distortion, intuitions from the solved Gaussian case are guaranteed to be relevant. This is because Shannon's upper and lower bounds for the rate–distortion function [11] can be extended to MD rate–distortion region bounds [162]. In any case, our interest is in the application of MD codes and a design methodology for one class of MD codes.

7.1.1 History

Goyal [69] reduces the history of MD coding to one sentence:

> "MD coding has come full circle from explicit practical motivation to theoretical novelty and back to engineering application."

The original motivation was to improve the reliability of the telephone system. By splitting the information of one connection across two separate links or paths and making do when one of the two is broken, one could improve reliability without having standby links. This idea was apparently conceived in the late 1970s amongst engineers at Bell Labs who worked on the physical layer [22, 112, 113]. It then passed within Bell Labs to speech coding researchers [62, 86, 87, 125, 127] and information theorists [156, 157, 158, 159].

MD coding was a popular theoretical novelty in the information theory community in the early 1980s, especially in the form of the "minimal breakdown degradation" problem [156]. In this problem one considers the coding of a memoryless binary symmetric source subject to Hamming distortion, with $R_1 = R_2 = \frac{1}{2}$ and $D_0 = 0$. The question: What is the minimum possible value for $D_1 = D_2$? Since the total rate of 1 bit/sample is precisely the minimum needed to achieve $D_0 = 0$, there is no excess rate to use to reduce the side distortions and seemingly little room for creative code design. Doing nothing special—splitting the source bits across the two descriptions—results in $D_1 = D_2 = 0.25$. Surprisingly, $D_1 = D_2 = (\sqrt{2} - 1)/2 \approx 0.207$ is achievable [49]. For quite some time the best lower bound was 0.2 [158]. The problem was ultimately resolved, with $(\sqrt{2}-1)/2$ established as the answer, by Berger and Zhang [12]. Remarkably, the full rate–distortion region for the case of a binary symmetric source with Hamming distortion is still not known. Only certain boundaries, such as the $R_1 + R_2 = 1$ boundary [2] and the $R_i = D(R_i)$, $i = 1, 2$, boundary [163] are known.

The development of practical MD coding techniques has been an active area of research for about ten years. The leading proponent was Vaishampayan, who, in some cases along with his students, developed fixed-rate [148] and variable-rate [151] MD scalar quantizers along with applications to speech communication [83, 147]. Vaishampayan advocated the use of MD coding for images and video [149], and the topic was picked up by many researchers. In recent years, several techniques based on lattice vector quantizers [153, 71], square transforms [150, 155, 72], and overcomplete transforms [73, 94] have been developed as abstract building blocks for various types of source data.

The current popularity of MD coding derives primarily from its applicability for packetized communications, such as with the Internet. In Internet communication, at least at the layers that are readily accessible, packets are either received correctly or lost completely. By associating packets with descriptions, MD coding is one way to mitigate the effect of packet losses. The alternatives to MD coding are to retransmit lost packets, such as with TCP [142], or to use conventional channel coding, referred to as forward error correction (FEC).

Each alternative is effective in some situations. Retransmission is perfectly efficient in the amount of received data (neglecting packet headers, the capacity of an erasure channel is achieved [41]) and is attractive when the delay induced by inferring a loss and retransmitting the lost packet is not prohibitive. Conventional FEC requires long block lengths and a well-chosen code rate, or a rateless code [19, 105], to be efficient in the amount of received data. Furthermore, most received packets are not useful until the end of a code block. Thus, retransmission and FEC require a feedback channel and/or long delay.

The utility of MD coding is when delay must be kept low (*e.g.*, for interactivity or streaming) and when a high data rate make buffering impractical (*e.g.*, for high-quality video). The goal of interactivity has prompted work on MD speech and audio coding [7, 89, 96, 133]. Recently, video coding has been the most active application area for MD [99, 134, 68, 82]. It is notable that the simple techniques of separating odd- and even-numbered video frames and of separating odd- and even-numbered scan lines are both effective, especially in conjunction with optimal mode selection [82]. These techniques are obvious analogies of the first MD coding technique for audio proposed by Jayant [86]. The following sections describe MD audio coding using odd-even separation and give an LMI-based design method for its optimization.

7.1.2 Comparison to Robust Predictive Quantization

Before describing the LMI approach to MD predictive coding, it is useful to understand when MD coding is useful in comparison to the robust predictive quantization described in Chapter 6.

The robust predictive quantization method described in the previous chapter applies best when the loss process acts on individual samples. This model, however, may not be well-suited for analyzing losses in packetized transmission. Specifically, predictive quantization often results in a small number of bits per sample. Consequently, to minimize the packetization overhead, a large number of quantized samples may be transmitted together. If the packet lengths are longer than the signal time constants, then it is impossible to estimate the lost data via interpolation, and robust predictive quantization will be ineffective.

As an example, consider packet transmission of speech. Suppose that each speech packet contain 40 ms of data, which is a typical number in interactive applications. Since much of the power spectral density of speech is above 25 Hz (1/40 ms), there will be minimal linear correlation between the data in different speech packets. Consequently, however one attempts to design a predictive encoder, it will be impossible to recover the data in the middle of any lost packet with a linear decoder. For such scenarios, modifying the prediction filter for improved robustness will provide minimal benefit.

However, we will see that with the odd-even separating encoder considered here, robust predictive quantization can be used for reliable transmission with arbitrarily long packets.

7.2 The Odd-Even Separation Method

The first name for MD coding was *channel splitting*, and the idea was simply to split data across two channels. How can one accomplish this split? First, the granularity of the splitting must be matched properly to the encoding technique: If the data is encoded into codewords in the usual sense (see, for example, [41]), then splitting a codeword is a bad idea because reconstruction from part of a codeword can be arbitrarily bad. For simplicity, we henceforth assume that the granularity of any splitting is in integral numbers of source samples.

The easiest way to split source samples is to alternate them across the two channels, separating odd- and even-numbered samples. When the samples are independent, each side decoder reconstruction has absolutely no information about half of the samples. The side distortions are quite high (at least half the variance of the source sequence, so the SNR is at most 3 dB), so this technique is virtually useless. On the other hand, when the samples are not independent, either side decoder can estimate the unavailable samples from the available samples. For example, if the source is approximately low pass, then the two side reconstructions could be two slightly different bandlimited interpolations of subsampled versions of the source. Thus, both side reconstructions can be rather good.

Since descriptions are either received without error or lost completely in the MD model, samples within a description can be compressed together without any loss in robustness.

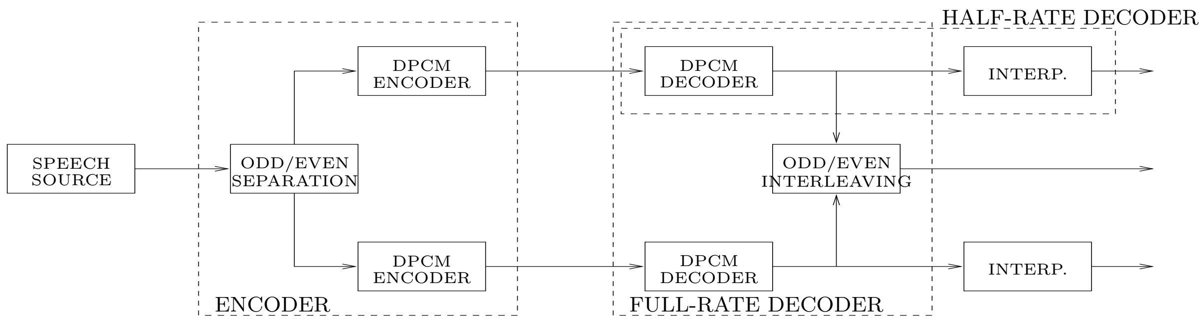

Figure 7.3. MD coding by odd-even separation and DPCM compression of each channel, as proposed by Jayant [86]. (Figure reproduced from [69].)

Combining this idea with odd-even separation, and considering a speech source, gives the technique of Jayant [86] depicted in Figure 7.3.[3] Each of the channels, at half of the original sample rate, is compressed by DPCM. Robustness is obtained because the interpolated signal obtained from either one description is close to the full-rate decoding. The redundancy between the two descriptions is due to the decrease in compressibility from keeping only half the samples. Note that as the compressibility of the source increases (spectral flatness of the source increases), the redundancy and the quality of the side reconstructions decrease together [69].

A similar but more general encoding structure is proposed in [83], as shown in Figure 7.4. The difference is that the odd-even split is performed on the quantized prediction errors rather than as the first step in the encoding. If the impulse response of the prediction filter is nonzero only at even times,[4] there is an equivalence to the previous system diagram. Appropriately choosing nonzero prediction filter values at odd times allows the overall encoding efficiency to be improved at the cost of higher side distortions—again, the standard MD trade-off.

Ingle and Vaishampayan [83] analyze and optimize the performance of the system in Figure 7.4 for a first-order Gauss-Markov source and a second-order predictor $P(z) = a_1 z^{-1} + a_2 z^{-2}$. They demonstrate that for the balanced case, a set of operating points can be achieved that dominate the performance of the earlier system. When applied to speech, under the simplifying assumption that the signal is first-order autoregressive with a correlation coefficient $r_{xx}(1)$ measured on a per-packet basis, the performance of this MD DPCM system is good.

[3]The encoding structure is nearly the same in [86] and [87]. The latter paper considers the performance when samples from each sub-channel are packetized and put on the same channel, with i.i.d. packet losses.

[4]For computability of the feedback loop, the prediction filter also must be strictly causal.

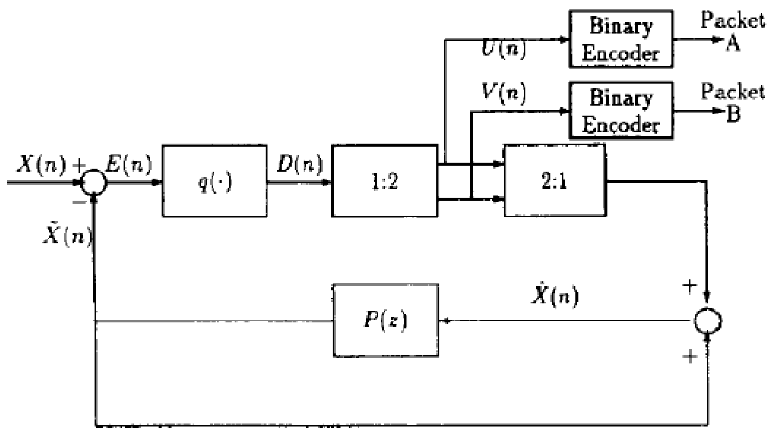

Figure 7.4. MD DPCM with odd-even separation as proposed by Ingle and Vaishampayan [83].

The main deficiency in the work on MD DPCM is the lack of a tractable design methodology. Even for the case of a first-order Gauss-Markov source and second-order predictor, the optimization of predictor gains a_1 and a_2 is difficult. In [83], the optimization is performed by searching over a discretized (a_1, a_2) plane. It is also proposed to make the search one-dimensional by restricting attention to (a_1, a_2) pairs that satisfy

$$\frac{a_1}{\rho} + \frac{a_2}{\rho^2} = 1, \qquad 0 \leq a_2 \leq \rho^2,$$

where ρ is the correlation coefficient in the AR source generation model. The intuition for this restriction is that it makes alternate samples of the prediction error sequence (neighboring samples on a single channel) uncorrelated, and the loss in performance as compared to optimizing over all (a_1, a_2) pairs is small. The use of a heuristic in even the first-order case highlights the need for an effective, general methodology.

7.3 Analysis and Design with LMIs

7.3.1 Odd-Even Separating MD Encoder

To analyze and design the odd-even separating coder using jump linear framework, we consider the MD coding system in Figure 7.5. As in Chapter 6, the source signal to be transmitted, $z[k]$, is modeled as the output of linear time-invariant state space system $H_z(z)$ driven by white noise $w_z[k]$. As usual, we will assume $H_z(z^{-1})$ admits a state space

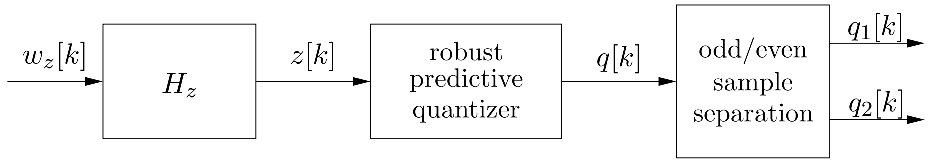

Figure 7.5. LMI design of odd-even sample separation MD coder: The source $z[k]$ is modeled as the output of a state space system $H_z(z)$ and is passed through a robust predictive quantizer to yield quantized samples $q[k]$. The samples $q[k]$ are split into the odd and even samples, $q_1[k]$ and $q_2[k]$, to obtain two descriptions, each at half the rate of $q[k]$.

realization of the form,

$$\begin{aligned} x[k+1] &= Ax[k] + Bw_z[k] \\ z[k] &= Cx[k] + Dw_z[k]. \end{aligned} \tag{7.2}$$

The source signal $z[k]$ is then quantized with the standard predictive quantizer we consider in Chapter 6,

$$\begin{aligned} x_E[k+1] &= Ax_E[k] + Lq[k] \\ z_E[k] &= Cx_E[k] \\ q[k] &= Q(z[k] - z_E[k]). \end{aligned} \tag{7.3}$$

As in Chapter 6, the predictive encoder (7.3) represents a predictive filter with output $z_E[k]$, in feedback with a scalar quantizer where $Q(\cdot)$. The prediction filter gain L is the free design parameter of the encoder.

The predictive coder yields a sequence of quantized samples $q[k]$. The two descriptions for the MD coder are created by simply splitting the sequence into odd and even samples defined by

$$q_1[k] = q[2k], \quad q_2[k] = q[2k+1].$$

The descriptions $q_1[k]$ and $q_2[k]$ each require half the bit rate of the samples $q[k]$. That is, if $Q(\cdot)$ outputs R bits per sample, each description, $q_1[k]$ and $q_2[k]$ will require $R/2$ bits per sample.

7.3.2 Jump Linear Analysis

Using the state space models for the signal and encoder, the performance of the MD encoder can be computed using the jump linear analysis in Chapter 6 as follows. As discussed earlier, the MD performance with two descriptions is essentially characterized by two metrics: (a) the central distortion, D_0, which is the achievable MSE in estimating $z[k]$ at the decoder when both descriptions are received; and (b) the side distortion, D_1, which is the MSE when only one of the two descriptions are received. Now, in the MD encoder in Figure 7.5, each description, $q_1[k]$ and $q_2[k]$, is simply some subset of the quantizer output samples $q[k]$. Thus, both the central and side distortions are precisely the MSEs in estimating $z[k]$ given some subset of the quantizer output samples $q[k]$: In the central distortion, the decoder receives all the samples, and in the side distortion, it receives every other sample.

In both of the cases, the set of samples that the decoder receives, can be described by a Markov chain driven erasure process. Specifically, let be $\theta[k]$ the random variable such that $\theta[k] = 0$ when the sample $q[k]$ is received by the decoder, and $\theta[k] = 1$ when it is not received. If both descriptions are received by the decoder, as in the central distortion case, $\theta[k] = 0$ for all k. In this case, $\theta[k]$ is a trivial Markov chain, with transition probabilities,

$$p_{00} = p_{10} = 1, \quad p_{01} = p_{11} = 0, \quad \text{(central distortion).} \tag{7.4}$$

That is, $\theta[k]$ always transitions to the non-lost state $\theta[k] = 0$. Alternatively, if the decoder receives only one description, $\theta[k]$ alternates between 0 and 1. This case can be described by the Markov chain with transition probabilities,

$$p_{01} = p_{10} = 1, \quad p_{00} = p_{11} = 0, \quad \text{(side distortion).} \tag{7.5}$$

Thus, the set of received samples can be described by an erasure process driven by one of two Markov chains transition probabilities: (7.4) for the case of receiving both descriptions, and (7.5) for the case of receiving only one description. We can then apply the LMI analysis method, in Algorithm 6.1 of the previous chapter with the two transition probabilities to determine the corresponding central and side distortions.

Using this analysis, the resulting decoder will be a jump linear estimator (6.19) with gain matrices L_{i1} and L_{i2} for $i = 0, 1$. Using the two transition probabilities will yield two sets of gain matrices: one set for the central decoder, and another for the side decoder.

We summarize the analysis procedure here:

Algorithm 7.1 (MD coder LMI analysis) *Consider the MD coder in Figure 7.5. Suppose the source signal $z[k]$ is generated by a stable LTI system of the form (7.2), the robust predictive coder is of the form (7.3) for a given encoder gain matrix L, and the quantizer can be described by (6.7). Let $\beta \in [0, 1]$ be the quantization coding loss, so that $\rho = 1 - \beta$. The central and side distortions can be computed as follows:*

1. *Use Proposition 6.1 to determine if the encoder gain matrix L is feasible. Specifically, verify that there is a solution $Q \geq 0$ to the Lyapunov equation (6.12), and confirm that the solution satisfies (6.13).*

2. *If the gain matrix L is feasible, compute the closed-loop quantization noise, σ_q^2 in (6.14).*

3. *For $i = 0, 1$, compute the jump linear system matrices A_i, B_i, C_{i1}, C_{i2}, D_{i1} and D_{i2} as in Section 6.3.2 where the state $\theta[k] = 0$ represent the non-loss state, and $\theta[k] = 1$ represents the loss state. That is, $I_{loss} = \{1\}$.*

4. *For the central distortion, use Theorem 3.2 with the transition probabilities (7.4) to compute the optimal gain matrices L_{i1} and L_{i2} for the estimator (6.19) for the central decoder. The resulting MSE in the theorem is the central distortion D_0.*

5. *For the side distortion, apply Theorem 3.2 again, but with the transition probabilities (7.5). This yields the optimal gain matrices L_{i1} and L_{i2} for the estimator (6.19) for the side decoder, and the resulting MSE is the side distortion D_1.*

The above algorithm gives a simple LMI procedure for computing the central and side distortions for a given encoder matrix L. However, this analysis makes two implicit assumptions. Firstly, for both D_0 and D_1, the algorithm computes only the *steady-state* distortion. The steady-state distortion is only relevant when the packets in each description are relatively long in comparison to the time constants of the signal and encoder. Under this assumption, when any packet is lost or received, we can assume that the decoder will eventually achieve close to the steady-state estimation performance within the decoding of that packet. If there is significant correlations of the signal between packets, the steady-state analysis will not apply.

Secondly, as in Chapter 6, we have restricted our attention to jump linear estimators of the form (6.19). The analysis does not consider the theoretically optimal time-varying Kalman filter, that may achieve a slightly lower distortion, especially in the transient regions at the packet boundaries. Also, the analysis does not consider nonlinear decoding techniques, such as those suggested in [138].

One other point needs some attention. Theorem 3.2 technically requires that the Markov chain $\theta[k]$ is aperiodic and irreducible. However, neither of the transition probabilities in (7.4) or (7.5) results in aperiodic, irreducible Markov chains. However, both cases have a simple solution.

For the central distortion case, the Markov chain $\theta[k] = 0$ for all k. To apply Theorem 3.2, we can thus eliminate the second state, $\theta[k] = 1$, to reduce to the trivial single state Markov chain. The constant Markov chain is equivalent to an LTI system, and can thus be analyzed with the standard LTI LMI theorem, Theorem 2.4 instead of the jump linear theorem 3.2.

For the side distortion, the Markov chain $\theta[k]$ is periodic and again Theorem 3.2 does not directly apply. To overcome this problem, we can analyze the side distortion by slightly perturbing the transition probabilities in (7.5) to obtain,

$$p_{01} = p_{10} = 1 - \epsilon, \quad p_{00} = p_{11} = \epsilon,$$

for some small $\epsilon > 0$. For any $\epsilon > 0$, the transition probabilities will no longer be periodic, and for small ϵ, the analysis will give a result very close to the true side distortion.

7.3.3 Encoder Gain Optimization

The analysis in Algorithm 7.1 computes the central distortion, D_0, and side distortion, D_1, for a *given* encoder gain matrix L. Mathematically, we could say that, for each gain matrix L, we obtain a distortion pair $(D_0(L), D_1(L))$. Searching over possible gain matrices L, would result in a set of achievable distortion pairs. One can then select the optimal distortion pair depending on the specifics of the problem. For example, one might wish to minimize some linear combination of the distortions $D_0(L)$ and $D_1(L)$. Alternatively, one could minimize one distortion subject to a hard constraint on the other.

However, whatever optimization criteria is used, it is not, in general, possible to exhaustively search over all possible gain matrices L. As discussed in Chapter 6 the distortions $D_0(L)$ and $D_1(L)$ will be complex non-convex functions of the gain matrix L.

To overcome this problem, we can use the suboptimal line search in Section 6.4. Specifically, for each $\lambda \in [0, 1]$, we can compute the distortion for $L = L(\lambda)$ given in (6.22). As λ varies from 0 to 1, we obtain a set of achievable distortion pairs $(D_0(L(\lambda)), D_1(L(\lambda)))$.

For the MD coder problem, the line search over λ has a simple intuitive appeal. When $\lambda = 1$, the encoder gain matrix $L(\lambda)$ is optimized for no loss. In the context of the MD

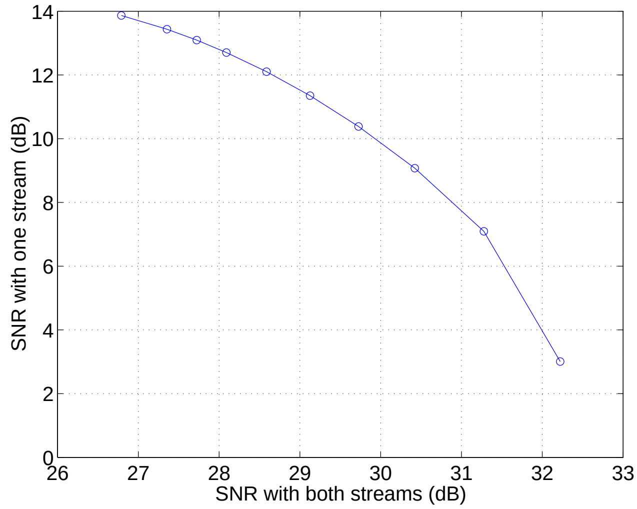

Figure 7.6. MD coding of a lowpass Gaussian signal. Plotted is the set of achievable central and side reconstruction SNRs using the LMI design method. The signal is a lowpass second-order ARMA Gaussian random process whose bandwidth is 0.1 times the sample rate. The encoder uses uniform scalar quantization with 3 bits per sample.

problem, this is equivalent to minimizing the central distortion D_0. As λ is decreased, the encoder is made more robust and consequently, the side distortion, D_1, is likely to improve at the expense of the central distortion, D_0. In this way, the line search provides a simple single parameter search for trading off the central and side distortions.

7.4 Numerical Example

As a simple numerical experiment, we return to the quantization of the lowpass Gaussian signal described in Section 5.7. In that example, the source $z[k]$ is modeled as the output of a second-order filter with a bandwidth equal to 0.1 times the sample rate. We will consider the encoding of this signal using a uniform scalar quantizer with 3 bits per sample.

For this problem, we consider the set of candidate encoder gain matrices $L(\lambda)$ as described in Section 7.3.3. For each λ, we compute the central and side distortions using Algorithm 7.1. The results are plotted in Figure 7.6.

Each point in the figure represents the distortion pair $(D_0(L), D_1(L))$ for some encoder gain matrix L. As the gain parameter λ varies from 0 to 1, we trace the curve of achievable distortion pairs for the matrices $L = L(\lambda)$. In the figure, the central and side distortions,

D_0 and D_1, in terms of the corresponding reconstruction SNRs:

$$\begin{aligned} \text{SNR (2 streams)} &= 10\log_{10}\left(\frac{\mathbf{E}|z[k]|^2}{D_0}\right). \\ \text{SNR (1 stream)} &= 10\log_{10}\left(\frac{\mathbf{E}|z[k]|^2}{D_1}\right). \end{aligned}$$

To understand the figure, we begin with the point at the bottom right of the curve. This point represents the case where the encoder is optimized for no loss. That is, the central distortion, D_0 is minimized. This point achieves the maximum reconstruction SNR when both streams are received. The reconstruction SNR is in excess of 32 dB. However, if only one stream is received, and we use this encoder the reconstruction SNR is very poor, approximately 3 dB.

What the figure shows is that the robustness with a loss of one stream can be greatly improved with a small sacrifice in the performance with both streams. For example, if we are willing to reduce the central reconstruction SNR by approximately 3 dB to 29 dB, we can improve the side reconstruction from approximately 3 dB to 11 dB. This demonstrates the principle of MD coding, with a relatively small loss in performance in the lossless case, we can obtain a significant improvement in the robustness to loss.

7.5 MD Audio Coding

The robust predictive quantization method described above is based on a state space model for the source. However, audio predictive quantization is rarely done with state space models, and is instead most often based on autoregressive (AR) signal models that are periodically re-estimated. Such adaptive AR models are well-suited for real-time audio applications since the AR modeling parameters can be easily estimated from the source signal, and the predictive encoder and decoder filters can, in turn, be easily computed from the AR model parameters.

In this section, we will show that the basic principles of the proposed MD predictive quantization described above can be applied to the standard AR approach as well. The resulting AR MD predictive encoder does not directly use the LMI optimization, but instead employs a simple heuristic method that is qualitatively similar and maintains the simplicity of the standard AR audio predictive approach.

7.5.1 AR Models and the Yule-Walker Equations

Suppose $z[k]$ is a wide-sense stationary random process. An *autoregressive* (AR) model for $z[k]$ is a recursive description of the form,

$$z[k] = \sum_{i=1}^{N} \alpha_i z[k-i] + w[k], \tag{7.6}$$

where $w[k]$ is some zero-mean white noise process and the coefficients α_i, $i = 1, \ldots, N$, are parameters that describe the AR model. In the frequency domain, the AR model can be written as $z = H_z(z^{-1})w$, where $H_z(z^{-1})$ is an IIR filter with the transfer function,

$$H_z(z^{-1}) = \frac{1}{1 - \sum_{i=1}^{N} \alpha_i z^{-i}}. \tag{7.7}$$

In the state space approach we have described so far, we have assumed that a state space model of the signal is given. However, in many circumstances, particularly in speech, the signal model must be estimated in real-time from the data. One of the appealing features of the AR model (7.6) is that the model parameters α_i can be easily estimated from data with a well-known technique called the *Yule-Walker* method.

The Yule-Walker method can be described as follows. Suppose we are given K samples of data, $z[k]$, $k = 0, \ldots, K-1$, and we wish to fit $z[k]$ with an AR model of the form (7.6). Assume that $z[k]$ is a zero-mean, wide sense stationary random process, with an autocorrelation function $R[j]$,

$$R[j] = \mathbf{E}(z[k]z[k-j]').$$

One can easily compute an estimate $\hat{R}[k]$ of the autocorrelation function from the sample mean,

$$\hat{R}[j] = \frac{1}{K-N} \sum_{k=N}^{K-1} z[k]z[k-j]'. \tag{7.8}$$

Now, using (7.6), one can show, for all $j = 1, \ldots, N$, the autocorrelation function satisfies

$$R[j] = \sum_{i=1}^{N} \alpha_i R[j-i]. \tag{7.9}$$

Thus, if the autocorrelation function, $R[j]$, were known, one can find the N AR model parameters, α_i, $i = 1, \ldots, N$, by solving a set of N linear equations. However, since the autocorrelation function is not exactly known, one can substitute the estimates $\hat{R}[j]$ for $R[j]$ in (7.9), and then solve the approximate equations to obtain estimates for the AR parameters α_i.

The equations (7.9) are known as the *Yule-Walker* equations. Due to their specific structure, the equations can be solved extremely efficiently through the well-known Levinson-Durbin algorithm. The overall procedure is thus very simple: take K samples of the data $z[k]$, compute the sample estimates, $\hat{R}[j]$, of the autocorrelation function, and then solve the Yule-Walker equations to obtain estimates of the AR parameters α_i.

7.5.2 Standard AR Encoder and Decoder

In addition to the simple model estimation, a second appealing feature of the AR model (7.6) is that optimal encoder and decoder filters have simple expressions in terms of the AR model parameters, assuming the quantization rate is high. Again, this material is standard and we will follow the presentation in [64].

Let $z[k]$ be the sequence to predictively quantized, and suppose that $z[k]$ can be modeled as an AR process of the form (7.6) for some coefficients α_i. The coefficients α_i could have been estimated from the Yule-Walker method. Let $z[k|k-1]$ be the one-step ahead predictor of $z[k]$ given past values of $z[j]$ up to the time $j = k-1$. That is,

$$z[k|k-1] = \mathbf{E}(z[k] \mid z[k-1], z[k-2], \ldots).$$

Taking the conditional expectation of both sides (7.6) with respect to the past samples of $z[k]$, we obtain the simple relationship

$$\begin{aligned}
\hat{z}[k|k-1] &= \mathbf{E}\left(z[k] \mid z[k-1], z[k-2], \ldots\right) \\
&= \sum_{j=1}^{K} \alpha_j z[k-j] + \mathbf{E}\left(w[k] \mid z[k-1], z[k-2], \ldots\right) \\
&= \sum_{j=1}^{K} \alpha_j z[k-j]
\end{aligned}$$

where, in the last step, we have used the fact that $w[k]$ is white, and thus uncorrelated with values of $z[j]$ for $j < k$. Therefore, the one-step ahead predictor is described by $\hat{z}[k|k-1] = H_{\text{pred}}(z^{-1})z[k]$, where $H_{\text{pred}}(z^{-1})$ is the FIR filter,

$$H_{\text{pred}}(z^{-1}) = \sum_{i=1}^{K} \alpha_i z^{-i}.$$

Observe that H_z in (7.7) is given by $H_z = (1 - H_{\text{pred}})^{-1}$. Using this fact along with the formulae in Section 5.6, we see that the optimal encoder and decoder filters for high rate are described by the transfer functions,

$$H_{\text{enc}}(z^{-1}) = \frac{H_{\text{pred}}(z^{-1})}{1 - H_{\text{pred}}(z^{-1})} = H_z(z^{-1}) - 1, \tag{7.10}$$

$$H_{\text{dec}}(z^{-1}) = \frac{1}{1 - H_{\text{pred}}(z^{-1})} = H_z(z^{-1}). \tag{7.11}$$

Now, recall that the encoder is constructed as a feedback of the quantizer $Q(\cdot)$ with the encoder filter, $H_{\text{enc}}(z^{-1})$. Thus, the encoder output $q[k]$ can be mathematically described by the feedback equations,

$$\begin{aligned} q[k] &= Q(z[k] - z_E[k]) \\ z_E[k] &= H_{\text{enc}}(z^{-1})q[k], \end{aligned} \tag{7.12}$$

where $z_E[k]$ is the encoder filter output. If we substitute the expression for $H_{\text{enc}}(z^{-1})$ in (7.10) into (7.12), and let $z_{E0} = (1 - H_{\text{pred}})^{-1}q$, we obtain the time domain equations,

$$\begin{aligned} q[k] &= Q(z[k] - z_E[k]) \\ z_E[k] &= \textstyle\sum_{i=1}^{N} \alpha_i z_{E0}[k-i] \\ z_{E0}[k] &= \textstyle\sum_{i=1}^{N} \alpha_i z_{E0}[k-i] + q[k]. \end{aligned} \tag{7.13}$$

Thus, we obtain a simple recursive set of equations for implementing the encoder. The coefficients, α_i, for the filters for the encoder are identical to those obtained in the AR model.

The equations for the decoder are also simple to derive. The decoder output, $z_D[k]$, is given by $z_D[k] = H_{\text{dec}}(z^{-1})q[k]$. Using the expression (7.11) for the decoder filter H_{dec}, the decoder output can be written as

$$z_D[k] = \sum_{i=1}^{N} \alpha_i z_D[k-i] + q[k]. \tag{7.14}$$

Therefore, the decoder can be written as a simple IIR filter whose coefficients, like the encoder filter, are given by the AR model. Observe that the update equation for $z_{E0}[k]$ in (7.13) and $z_D[k]$ in (7.14) are identical. Thus, assuming the AR filter H_{pred} is stable, $z_{E0}[k] - z_D[k] \to 0$ as $k \to \infty$. Consequently, the decoder will asymptotically "know" the encoder state $z_{E0}[k]$.

Many audio applications use the above procedure with an important modification. Audio signals are often not well-modeled as a random process whose statistics are stationary over a long period. Instead, it is useful to periodically re-estimate the AR parameters. To this end, the signal is usually divided into small intervals, typically in the tens of milliseconds. The above procedure including the model estimation, encoding and decoding is then repeated in each interval. The resulting quantized data will thus include both the sequence of quantized predictive encoder outputs along with periodic updates on the AR model parameters.

The overall AR predictive quantization can be summarized as follows: Let K be the block length, and N be the AR model order.

- Let $z[k]$, $k = 0, \ldots, K-1$ by the samples to be quantized. The samples could represent the entire sequence, or as in audio applications, a K-length block of samples over which the AR model is to be fit.
- Obtain estimates $\hat{R}[j]$ as in (7.8), $j = -N+1$ to $j = N$ for the autocorrelation function $R[j]$.
- Estimate the coefficients α_i, $i = 1, \ldots, N$ in the AR model (7.6) by solving the Yule-Walker equations (7.9) with the estimate $\hat{R}[j]$ replacing $R[j]$. The equations can be solved via the standard Levinson-Durbin algorithm.
- Predictively encode $z[k]$ with the equations (7.13).
- Transmit the quantized values $q[k]$ *and* the AR model coefficients α_i to the decoder.
- The decoder constructs the signal estimate $z_D[k]$ from the IIR filter (7.14).
- The procedure is repeated for each block of K samples.

7.5.3 Proposed MD Audio Predictive Coder

In principle, the techniques in Section 7.3 can be directly applied to the standard AR predictive encoder described above to obtain an MD audio predictive encoder. As before, in each interval of K samples of $z[k]$, one would first estimate the AR model (7.6) with the standard Yule-Walker method. However, instead of using the corresponding AR encoder filter in (7.10), one would use a robust encoder filter as designed as in Section 7.3. The output samples of the robust predictive encoder would then be divided into two descriptions. Depending on which descriptions arrive, the decoder can then apply an appropriate jump linear decoder designed by an LMI.

Unfortunately, while this technique is theoretically feasible, it is computationally expensive for audio applications. In each interval, the encoder would have to transform the AR signal model into state space and then solve a modified Riccati equation to obtain the robust predictor gain L. Since the intervals are typically short (in the tens of milliseconds), this is an extremely expensive operation to perform frequently. Also, for the decoder gains, it is not practical to solve an LMI in each coding interval.

We thus consider an alternative, simpler heuristic encoder technique that is, qualitatively similar to the LMI-based method. As before, let $z[k]$, $k = 0, \ldots, K-1$ be the K samples of an interval of audio signal to encode. Also as before, compute the estimates $\hat{R}[j]$ in (7.8) of the autocorrelation function, and solve the Yule-Walker equations,

$$\hat{R}[j] = \sum_{i=1}^{N} \alpha_i \hat{R}[j-i], \quad j = 1, \ldots, N, \tag{7.15}$$

to obtain estimates, α_i, for the AR model of $z[k]$.

Now let $\lambda \in [0, 1]$. The variable λ is a robustness parameter that will play a similar role as in the encoder gain matrix design in Section 7.3.3. Given λ, let $\hat{R}_\lambda[j]$ be the *modified* autocorrelation function estimate

$$\hat{R}_\lambda[j] = \begin{cases} \hat{R}[j] & \text{if } j \text{ is even,} \\ \lambda \hat{R}[j] & \text{if } j \text{ is odd.} \end{cases} \tag{7.16}$$

Let $\alpha_i(\lambda)$ be the solution to the corresponding Yule-Walker equations,

$$\hat{R}_\lambda[j] = \sum_{i=1}^{N} \alpha_i(\lambda) \hat{R}_\lambda[j-i]. \tag{7.17}$$

For the robust encoder, we propose to replace the AR coefficients α_i in (7.13) with the modified coefficients $\alpha_i(\lambda)$ to obtain the predictive encoder,

$$\begin{aligned} q[k] &= Q(z[k] - z_E[k]) \\ z_E[k] &= \textstyle\sum_{i=1}^{N} \alpha_i(\lambda) z_{E0}[k-i] \\ z_{E0}[k] &= \textstyle\sum_{i=1}^{N} \alpha_i(\lambda) z_{E0}[k-i] + q[k]. \end{aligned} \tag{7.18}$$

The resulting quantizer output samples are then divided into the two descriptions: the even samples, $q[2k]$, and the odd samples, $q[2k+1]$.

The encoder in (7.18) is motivated by similar reasoning as the MD encoder in Section 7.3.3. First consider the case when $\lambda = 1$. In this case, the modified autocorrelation estimate, $\hat{R}_\lambda[j]$ in (7.16), reduces to the standard autocorrelation estimate $\hat{R}[j]$. Therefore, the parameters $\alpha_i(\lambda)$ and the encoder (7.18) reduce to the standard AR parameters α_i and AR encoder (7.13), respectively. Thus, when $\lambda = 1$, the encoder is identical to the standard AR design which assumes that all the samples are received. In this sense, setting $\lambda = 1$

optimizes the encoder for no loss – that is, both descriptions are arrive at the decoder. In the context of the MD system, selecting $\lambda = 1$ will thus attempt to minimize the central distortion.

Now consider the case when $\lambda < 1$. From (7.16), we see that the samples $\hat{R}_\lambda[j]$ are reduced for odd samples j. When the values $\hat{R}_\lambda[j]$ are used in the Yule-Walker equations (7.17), the resulting prediction coefficients α_i will therefore naturally discount the correlation from the current sample to any sample an odd number of samples in the past. The motivation is clear: if only one description is received at the decoder, the decoder will receive only every other sample of the quantizer. Thus, when only one description is present, on any sample that is received, all samples that are an odd number of samples in the past will be missing. To improve the robustness of the encoder, the prediction filter should reduce the weight of these samples in the prediction.

In the limit, as $\lambda \to 0$, it is easy to show that

$$\lim_{\lambda \to 0} \alpha_i(\lambda) = 0, \quad \text{for all } i \text{ odd.}$$

Thus, as $\lambda \to 0$, the prediction filter coefficients completely ignores all samples an odd number of samples in the past, and encode as if only every other sample is received. In this sense, the encoder becomes optimized for the case of one stream being received, and will therefore, presumably minimize the side distortion.

We therefore see that the parameter λ provides a simple tunable knob to trade off the central and side distortion. As $\lambda \to 1$, the encoder will be optimized for the central distortion (no loss), and as $\lambda \to 0$, the encoder will minimize the side distortion (every other sample received).

The encoder procedure can be summarized as follows. There are three parameters: the block length K, the AR model order M and a robustness parameter $\lambda \in [0, 1]$. The audio signal can then be encoded with the following procedure:

- Let $z[k]$, $k = 0, \ldots, K-1$ by the samples to be quantized. The samples could represent the entire sequence, or as in audio applications, a K-length block of samples over which the AR model is to be fit.
- Obtain estimates $\hat{R}[j]$ as in (7.8), $j = -N + 1$ to $j = N$ for the autocorrelation function $R[j]$.
- Estimate the coefficients α_i, $i = 1, \ldots, N$ in the AR model (7.6) by solving the Yule-

Walker equations (7.15). The equations can be solved via the standard Levinson-Durbin algorithm.

- Also compute the modified AR coefficients $\alpha_i(\lambda)$, by solving the Yule-Walker equations (7.17).
- Generate the K quantized samples $q[k]$ with the encoder (7.18).
- Form one description with the odd samples $q[2k]$ and a second description with the odd samples $q[2k+1]$. In *both* descriptions include the AR parameters $\alpha_i(\lambda)$ and α_i.
- The procedure is repeated for each block of K samples.

For the decoder, it is not practical to use an LMI in each coding block to compute the gains for the jump linear estimator. Since the block lengths are typically small, it is simpler to employ the optimal time-varying Kalman filter. As we have discussed earlier, the Kalman filter requires states for both the signal and encoder. Given an AR signal model (7.6) and predictive encoder (7.18), one possible state vector is the $2M$-dimension state,

$$x[k] = [z[k], \ldots, z[k-M+1], z_{E0}[k], \ldots, z_{E0}[k-M+1]] .$$

Here, the first M state components are for the signal $z[k]$, and the second M components are for the encoder filter. Using the AR signal model (7.6) and encoder filter (7.18), one can construct the state space matrices for $x[k]$, and then apply the Kalman filter to estimate $z[k]$ from whatever samples $q[k]$ arrive.

7.5.4 Audio Example

To illustrate the above technique, we apply the MD encoder to a set of audio samples from the Sound Quality Assessment Material, or SQAM CD [54]. The SQAM files are industry standard audio files, primarily used in speech compression evaluation. In [54], each audio sample is stored in the RIFF WAVE standard with 16-bit stereo data sampled at 44.1 kHz. For the testing, we converted the samples to mono, and then downsampled by a factor of four, to obtain test files at 11.025 kHz. The proposed audio MD decoder was then performed with an $M = 4$ order AR encoder, and 40 ms blocks. At the 11.025 kHz samples, each block contains 441 samples. The quantizer uses 6 bits per sample uniform quantization to give a rate of 66.1 kbps, or 33.1 kbps per description.

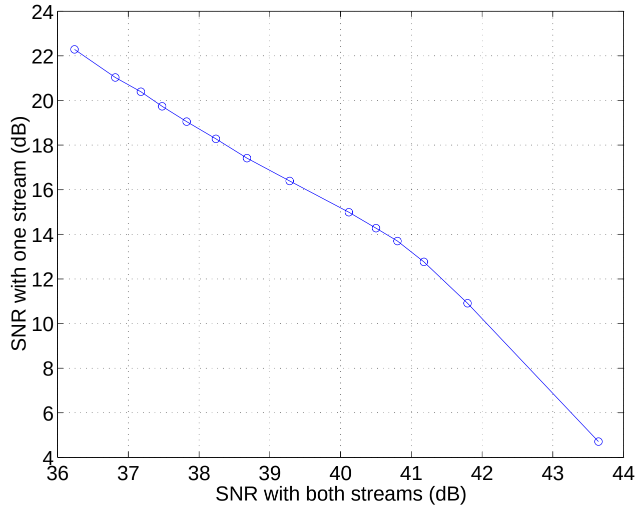

Figure 7.7. MD coding of an audio signal `vioo10_2.wav` from [54]. The MD procedure is performed on a mono signal at 11.0 kHz sampling, with fourth-order AR model and 6 bits per sample uniform quantization. The total rate is 66.1 kbps, or 33.1 kbps in each description. Plotted is the set of central and side reconstruction SNRs at various robustness levels λ. Observe that a small degradation in the central SNR can result in a large improvement in the side reconstruction.

This rate does not include the rate for encoding the AR coefficients α_i and $\alpha_i(\lambda)$. However, with an $M = 4$ order AR model, there are only $2M = 8$ such coefficients per block of 441 samples. Consequently, the overhead in transmitting the AR model coefficients will be relatively small.

The results of encoding a 10 second excerpt of the audio file, `vioo10_2.wav`, is shown in Figure 7.7. Each point in the encoder represents the performance of the MD encoder for one particular value of the robustness parameter λ. The curve is obtained by testing the encoder with various parameter values from $\lambda = 0$ to 1. For each parameter value, the figure plots the reconstruction SNR when both descriptions are received, and when only one description is received.

The point at the bottom right of the curve represents the point where $\lambda = 1$ and the MD encoder is optimized for no loss. This point maximizes the central SNR (the SNR when both descriptions arrive) and obtains a value of 43.6 dB. However, if one of the descriptions are lost, the decoder obtains a very poor performance with an SNR of only 4.7 dB. Although this cannot be depicted in the figure, with a loss of one of the descriptions, there is a strong ringing sound in the reproduction at the decoder that is a result of the persistent effects of the losses in the predictive coder.

Encoder	**Reconstruction SNR (dB)**	
	Both streams	One stream
MD with $\lambda = 0$	36.2	22.3
MD with $\lambda = 0.82$	37.8	19.1
MD with $\lambda = 1$	43.6	4.7
Full rate, no protection	43.6	-
Half rate, repeated descriptions	27.2	27.2

Table 7.1. Numerical values for points in Figure 7.7. Also shown for comparison is the performance of optimal predictive encoding with no loss at the full rate of 66.1 kbps, and predictive encoding at half the rate of 33.1 kbps, with quantized values repeated in each description.

As λ is decreased, we move left in the curve and the reconstruction SNR with only one stream dramatically improves with only a small reduction in performance with two streams. For example, to obtain a 19.1 dB SNR with one stream, the two stream SNR only needs to decrease from 43.6 to 37.8 dB. Again, while the perception of the audio reconstruction cannot be shown in the figure, the two stream SNR degradation is hardly noticeable. However, for the one stream case, the ringing sound is virtually eliminated and the degradation appears as a small hiss. We therefore see the benefit of MD coding: for a small degradation in performance with no loss, a reasonable signal can be obtained when there is a loss.

Numerical values for a few of the points in Figure 7.7 are shown in Table 7.1. The tables shows the central and side reconstruction SNR for robustness parameters of $\lambda = 0$, 0.82 and 1. The table also compares these MD performance values to two simple single description alternatives: a single description predictive encoder designed at full rate assuming no loss; and a single description decoder designed at half the rate, with the encoder outputs repeated in each stream for protection against loss of one stream. To simplify the comparison, the single description designs were performed with the same $M = 4$ AR model order, 40ms block length and sample rate of 11.0 kHz. The full rate encoder uses 6 bits per sample uniform quantization rate and is thus identical to the MD encoder with $\lambda = 1$. Therefore, when there is no loss, the MD encoder and single description full rate encoder will obtain the same performance of 43.6 dB. However, when there is loss both the MD encoder with $\lambda = 1$ and the full rate encoder will perform poorly.

The half-rate encoder is run at 3 bits per sample, with the same sample rate of 11.0 kHz. This results in half the total rate, and consequently, the data can be repeated in each description. As shown in Table 7.1, this method results in a 27.2 dB reconstruction SNR.

Since the same data is repeated in each stream, the SNR is achievable even if only one stream arrives. However, if both streams arrive, there is no performance improvement. The full rate and half rate encoders thus represent two extremes: one optimized for no loss, and the other for loss of one stream. In comparison, the MD encoder performance fall between these two encoders, with reasonable performance in either the loss and no loss case.

7.6 Sample Rate Optimization

An important issue for future study that should be mentioned at this point is the question of the sample rate selection. We saw in Section 5.8, that in quantizing a correlated random process, there is an optimal sample rate that is a function of the signal power spectral density and quantization rate. As the quantization rate is increased, the optimal sampling rate increases to attempt to capture more of the signal energy.

In our description of the MD decoder up to now, we have assumed that the sample rate is essentially given. However, in practice, by appropriately downsampling the signal, the sample rate is free to be chosen. An interesting and important practical question then is what sample rate should be selected when using the proposed sample separating MD encoder?

To understand this problem, consider the proposed MD encoder operated at some sample rate f samples per second, and an overall quantization rate of R bits per second. Since the two descriptions in the MD encoder are formed from taking the odd and even samples of the encoder output, each description effectively operates half the sample rate, $f/2$, and carries half the information rate, $R/2$.

Now let $f_{\text{opt}}(R)$ be the optimal sample rate for standard (single description) predictive quantization. This optimal sampling rate could, theoretically, be determined as in Section 5.8. For the MD encoder, one would expect, that as one optimizes for the central distortion (i.e. all the samples are received), the encoder sample rate, f, should approach $f_{\text{opt}}(R)$. Similarly, as one optimizes for the side distortion (i.e. only one description is received), the encoder sample rate for each description, $f/2$, should approach $f_{\text{opt}}(R/2)$. Therefore f should approach $2f_{\text{opt}}(R/2)$. We thus see an interesting property: as one trades off the central vs. side distortion, the sample rate should be changed from $f_{\text{opt}}(R/2)$ to $2f_{\text{opt}}(R)$.

To get an idea of what this sample rate transition looks like, suppose $z[k]$ is a lowpass signal with a flat power spectral density, and a sharp cutoff frequency at f_c. Then, it is easy to verify that the optimal sample rate $f_{\text{opt}}(R) = f_c$ for all rates R. That is, for standard encoding, the signal $z[k]$ should be quantized exactly at the Nyquist rate of the signal. However, using the formula above, in the MD case, the encoder sample rate f should be varied from f_c to $2f_c$. The rate f should be set to f_c when optimizing for the central distortion, and $2f_c$ when optimizing for the side distortion. In other words, increasing the robustness of the MD encoder, requires oversampling the signal by a factor up to two.

Unfortunately, our experiments and analysis have not yet considered this sample rate optimization. Nevertheless, it will be an interesting avenue of future research.

Chapter 8

Conclusions

8.1 Summary

Our original problem was to understand how to trade off robustness and compression in communication of correlated signals without channel coding. Correlated signals inherently have redundancy that can either be removed for improved compression, or maintained for robustness to losses. To study this problem, we have proposed a jump linear model that can capture arbitrary second-order statistics of the signal, linear filters, and Markovian variations and losses in the channel. The model is extremely general and facilitates easy numerical analysis. The main limitation is that the decoder must be linear.

A natural performance metric is the minimum achievable error in reconstructing the transmitted signal at the decoder. Using the jump linear model, we have shown that this average signal reconstruction error can be bounded by an LMI optimization. The optimization provides an efficient numerical tool for evaluating a general, linear transmission in terms of the signal statistics, linear characteristics of the encoder and quantizer, and loss model.

The result is then applied to predictive quantization, which is a widely used technique for removing redundancy in signals to improve compression. While predictive quantization can offer significant gains in compression, it is well-known that the quantized data may not be robust to losses. The LMI analysis of the jump linear model provides a numerical method for precisely quantifying this performance degradation. A heuristic method for encoder design is proposed to improve the robustness. The proposed encoder allows a simple mechanism for trading off robustness and compression.

Finally, a multiple description (MD) coder is presented based on using the robust predictive quantization encoder along with separation of the output streams into odd- and even-numbered samples. By varying the robustness level of the encoder, one can trade off central and side distortion. The potential of this MD coding structure is demonstrated by implementing a block-adaptive audio coder.

8.2 Future Extensions

8.2.1 Estimation theory

While the jump linear estimation theory in Chapter 3 is the most settled portion of this work, there are, nevertheless, several remaining problems to consider.

One important open problem is to extend the jump linear analysis to the case of fixed-interval smoothing as described in Section 2.4.5. For lossy transmission, smoothing is particularly useful for recovering from losses, since the decoder can interpolate from samples both in the past and future. In Section 2.4.5, we saw that for an LTI system, the asymptotic fixed interval smoothing error is described by a solution to the standard algebraic Riccati equation for the forward filter, along with a Lyapunov equation for the reverse filter. For the jump linear case, one would expect that the error would be described by a set of coupled Riccati equations for the forward filter and coupled Lyapunov equations for the reverse filter.

Another open question is related to optimality. The LMI optimization in Chapter 3 provides an upper bound on the average estimation error. In certain examples in Section 4.7, we saw that the LMI upper bound was very close to the minimum error achieved by the Kalman filter. It would be interesting to investigate the tightness of this bound and find sets of extremal examples. Furthermore, it would be nice to find an LMI *lower* bound, especially as a means to understand the tightness of the upper bound.

8.2.2 Robust Predictive Quantization and MD Coding

While the robust predictive quantization results suggest the value of LMIs in analysis of predictive quantization subject to losses and MD coding, there is much to be understood. One of the large open questions is sample rate optimization. As discussed in Section 5.8, for lossless encoding, the optimal sample rate can be computed from rate–distortion theory. However, how to optimize the sample rate for losses is an open question. Presumably, if the quantization rate is high, then oversampling affords greater redundancy and may be useful in reducing the overall distortion. For the MD case, the sample rate optimization is further complicated by the fact that each stream has to run at half the sampling rate as the overall rate. Some of these issues were qualitatively discussed in Section 7.6, but extensions to the theory are needed to investigate this.

A second issue is the only-approximate optimality of the encoder design in both the robust predictive quantization and MD coding problems. The encoder design proposed in Chapter 6 and used also in Chapter 7 reduces a high-dimensional optimization to a search over a parameter $\lambda \in [0, 1]$. This is reasonable and effective, but not necessarily optimal. It would be interesting to compare the performance of this coder with an exhaustive search of encoder filters, at least for a low-order encoder. Even better would be an analytic method that provides an optimization over a larger class of filters.

A related question is the selection of the encoder filter order. In the current design, the encoder filter order is equal to the order of the signal model for $z[k]$. However, with losses there may be some benefit to increasing the order. One way a higher-order coder may arise naturally in the current framework is when the signal is filtered and then oversampled. The filtering will introduce further states into the signal model and thereby increase the order of the encoder filter.

Finally, it would be relatively easy to investigate the value of feedback. Specifically, suppose the encoder knew the channel state $\theta[k]$. With feedback, one can apply an identical argument as in Proposition 5.1 to show that the optimal encoder is given by

$$z_E[k] = \mathbf{E}\left(z[k] \mid \hat{q}[0:k-1]\right),$$

which is the optimal predictor of $z[k]$ given the values $\hat{q}[k]$ actually received at the decoder. Since $\hat{q}[k]$ is the output of a jump linear system, it is possible that the performance of this system could be studied with an LMI analysis.

8.2.3 Audio and Video Applications

The audio coding example presented in Section 7.5 is effective as a proof of concept. Although the proposed technique enables a relative trade-off between the central and side distortion, the performance in terms of absolute bit rate in either case is not impressive. Indeed, with 66 kbps, one can usually obtain much higher quality audio than realized in this experiment. An application better suited to predictive quantization would have been speech coding rather than audio coding.

There is great potential to use the techniques of this dissertation to analyze and optimize the performance of video coding systems used over lossy channels. Many key parameters, such as the frequency of I frames, are chosen in part based on propagation and persistence of errors. These parameters are currently tuned based on simulation rather than analysis. An analytical basis might allow more effective and efficient adaptation of parameters.

8.3 Other Connections

8.3.1 Uncoded Communication

Most of our results have concerned communication of quantized data over a erasure channels without channel coding. However, Gastpar *et al.* [59] have recently considered uncoded transmission in the sense of sending *unquantized* data over a continuous-valued channels with additive noise. The simplest case concerns the communication of a sequence of i.i.d. Gaussian random variables $z[k]$ over an AWGN channel with an average power constraint. The separation principle dictates that the optimal communication is given by optimal vector quantization of a large number of number of samples of $z[k]$ followed by an optimal channel code to mitigate the effect of the additive noise. However, for transmitting a Gaussian random variable over an AWGN channel, the same average distortion can be obtained by simply scaling the values $z[k]$ and transmitting the continuous-valued signal over the channel. Gastpar [59] characterizes other source and noise distributions where such uncoded, memoryless transmission is optimal.

Part of this dissertation can be interpreted as extending uncoded communication to correlated sources sent over channels with memory. This line of research may provide some insight in what role signal and channel bandwidth play in uncoded transmission.

8.3.2 Compression of Unstable Processes

Hashimoto and Arimoto [81] and more recently Sahai and Mitter [130] have considered the problem of compression of unstable random processes. The framework presented here may be useful for this problem as well. For example, in Chapter 5 it was assumed that the state space system (5.9) was stable, so that the resulting random process $z[k]$ has bounded variance. However, one could in principle consider an unstable system, but attempt to find a encoder–decoder pair with bounded error variance, $\mathbf{E}|z[k] - z_D[k]|^2$, even though the underlying random process $z[k]$ has unbounded variance. Under the assumption of the linear quantizer model, the analysis in Theorem 5.1 shows that the existence of a stabilizing coder is equivalent to the existence of a solution $P \geq 0$ to the modified algebraic Riccati equation (5.14). Now, we have mentioned earlier that this equation has been studied by Sinopoli *et al.* [140]. One of the results there is: if (A, B) is stabilizable, a necessary condition for there to exist a solution $P \geq 0$ to (5.14) is that

$$\beta \lambda_{\max}(A)^2 < 1,$$

where $\lambda_{\max}(A)$ is the maximum absolute eigenvalue of A. Using the rate–distortion bound that $\beta \geq 2^{-2R}$, we obtain

$$R \geq \log_2 \left(\lambda_{max}(A) \right), \tag{8.1}$$

which provides a simple lower bound on the rate required for compression with finite variance. Sahai and Mitter have derived a similar result, but restricted to an AR-1 random process with bounded noise. Of course, the analysis there is more exact; it does not assume the linear quantization model or Gaussian quantization error. Sahai and Mitter have also considered the effect of quantization losses. Nevertheless, there may be interesting connections between the two approaches.

8.3.3 Wyner–Ziv Coding

It is interesting to compare the robust predictive quantizer presented here with Wyner–Ziv coding [161] and the Wyner–Ziv video coders recently proposed by Puri, Ramchandran, and Majumdar [124, 107]. Wyner–Ziv coding allows the source correlation to be exploited entirely at the decoder rather than at the encoder. Comparable to this in the framework of this dissertation is to remove the encoder filter and do all the work at the decoder.

The Wyner–Ziv coder may use a memoryless coset code or some similar operation at the encoder and will have a highly nonlinear decoder. The robust predictive quantizer can be seen as a "poor man's Wyner–Ziv coder" because its decoder is constrained to be linear. Given the linear constraint on the decoder, it seems that the encoder should use a regular quantizer (i.e., a quantizer with convex partition cells), so the encoder is also simple.

8.3.4 Applications to Finite Bit Rate Control

We have seen that there is a fundamental duality between state estimation and state-feedback control. In this context, the natural control analogue of the predictive quantization problem is the problem of finite bit rate state-feedback control. This problem has been recently studied by several researchers including [50, 115, 129, 146, 145]. Nair and Evans [115] have shown that, under very general conditions, a linear state space system is stabilizable with a finite bit rate R, if and only if

$$R \geq \sum_{i=1}^{M} \log_2(\mu_i),$$

where the sum is over all the unstable eigenvalues μ_i of A. The bound is similar to the bound (8.1) for the compression of unstable ARMA processes.

However, while the Nair and Evans result precisely describes when a system is stabilizable with finite bit rate, the actual achievable steady-state variance is not computed. It would be interesting to apply the state space framework of Chapter 5 to this problem to quantify the steady-state error with finite bit rate control. Using a jump linear model, this analysis may also be able to consider the effect of losses.

Bibliography

[1] G. A. Ackerson and K. S. Fu. On state estimation in switching environments. *IEEE Trans. Automat. Control*, AC-15(1):10–15, February 1970.

[2] R. Ahlswede. The rate distortion region for multiple descriptions without excess rate. *IEEE Trans. Inform. Theory*, IT-31(6):721–726, November 1985.

[3] M. Ait Rami and L. El Ghaoui. LMI optimization for nonstandard Riccati equations arising in stochastic control. *IEEE Trans. Automat. Control*, 41(11):1666–1671, November 1996.

[4] R. Akella and P. R. Kumar. Optimal control of production rate in a failure prone manufacturing system. *IEEE Trans. Automat. Control*, 31(2):116–126, February 1986.

[5] V. R. Algazi and J. T. DeWitte, Jr. Theoretical performance in entropy-encoded DPCM. *IEEE Trans. Comm.*, COM-30(5):1088–1095, May 1982.

[6] R. Anand, K. Ramchandran, and I. V. Kozintsev. Continuous error detection (CED) for reliable communication. *IEEE Trans. Comm.*, 49(9):1540–1549, September 2001.

[7] R. Arean, J. Kovačević, and V. K Goyal. Multiple description perceptual audio coding with correlating transforms. *IEEE Trans. Speech Audio Proc.*, 8(2):140–145, March 2000.

[8] D. S. Arnstein. Quantization error in predictive coders. *IEEE Trans. Comm.*, COM-23(4):423–429, April 1975.

[9] A. V. Balakrishnan. *Kalman Filtering Theory.* Springer, New York, February 1984.

[10] Y. Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Trans. Automat. Control*, AC-23(4):618–626, August 1978.

[11] T. Berger. *Rate Distortion Theory.* Prentice-Hall, Englewood Cliffs, NJ, 1971.

[12] T. Berger and Z. Zhang. Minimum breakdown degradation in binary source encoding. *IEEE Trans. Inform. Theory*, IT-29(6):807–814, November 1983.

[13] R. R. Bitmead and B. D. O. Anderson. Lyapunov techniques for the exponential stability of linear difference equations with random coefficients. *IEEE Trans. Automat. Control*, AC-25(4):782–787, August 1980.

[14] W. P. Blair and D. D. Sworder. Continuous time regulation of a class of econometric models. *IEEE Trans. on Syst. Man and Cybernetics*, SMC-5:341–346, 1975.

[15] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Trans. Automat. Control*, 33(8):780–783, August 1988.

[16] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia, PA, 1994.

[17] G. Buch, F. Burkert, J. Hagenauer, and B. Kukla. To compress or not to compress? In *IEEE Globecom*, pages 196–203, London, November 1996.

[18] J. A. Bucklew and N. C. Gallagher, Jr. Some properties of uniform step size quantizers. *IEEE Trans. Inform. Theory*, IT-26(5):610–613, September 1980.

[19] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In *Proc. ACM SIGCOMM*, pages 56–67, Vancouver, September 1998.

[20] P. E. Caines. *Linear Stochastic Systems.* John Wiley and Sons, New York, 1988.

[21] F. M. Callier and C. A. Desoer. *Linear Systems Theory.* Springer, New York, 1994.

[22] J. C. Candy, D. Gloge, D. J. Goodman, W. M. Hubbard, and K. Ogawa. Protection by diversity at reduced quality. Bell Laboratories Memorandum for Record (not archived), June 1978.

[23] Y.-L. Chen and B.-S. Chen. Model-based multirate representation of speech signals and its applications to recover of missing speech packets. *IEEE Trans. Speech Audio Proc.*, 5(3):220–231, May 1997.

[24] J. Chou and K. Ramchandran. Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission. *IEEE J. Sel. Areas Comm.*, 18(6):861–867, June 2000.

[25] P. M. Clarkson. *Optimal and Adaptive Signal Processing.* CRC Press, Boca Raton, FL, 1993.

[26] E. F. Costa and J. B. R. do Val. Full information H_∞ control for discrete-time infinite Markov jump parameter systems. *Automatica*, 202:578–603, 1996.

[27] E. F. Costa and J. B. R. do Val. On the detectability and observability of discrete-time Markov jump linear systems. *Systems and Control Letters*, 44:135–145, 2001.

[28] O. L. V. Costa. Linear minimum mean square error estimation for discrete-time Markovian jump linear systems. *IEEE Trans. Automat. Control*, 39(8):1685–1689, August 1994.

[29] O. L. V. Costa. Discrete-time coupled Riccati equations for systems with Markovian switching parameters. *J. Math. Anal. Appl.*, 194:197–216, 1995.

[30] O. L. V. Costa, E. O. Assumpcao, E. K. Boukas, and R. P. Marques. Constrained quadratic state-feedback control of discrete-time Markovian jump linear systems. *Automatica*, 35:617–626, 1999.

[31] O. L. V. Costa, J. B. R. do Val, and J. C. Geromel. A convex programming approach to H_2-control of discrete-time Markovian jump linear systems. *Int. J. Control*, 66(4):557–559, April 1997.

[32] O. L. V. Costa, J. B. R. do Val, and J. C. Geromel. Continuous-time state-feedback H_2-control of Markovian jump linear systems via convex analysis. *Automatica*, 35:259–268, 1999.

[33] O. L. V. Costa and M. D. Fragoso. On the existence of maximal solution for generalized algebraic Riccati equations arising in stochastic control. *Systems and Control Letters*, 14:233–239, 1990.

[34] O. L. V. Costa and M. D. Fragoso. Stability results for discrete-time linear systems with Markovian jumping parameters. *Int. J. Contr.*, 66:557–579, 1993.

[35] O. L. V. Costa, M. D. Fragoso, and R. P. Marques. *Discrete-Time Markov Jump Linear Systems*. Probability and Its Applications. Springer, London, 2005.

[36] O. L. V. Costa and S. Guerra. Robust linear filtering for discrete-time hybrid Markov linear systems. *Int. J. Control*, 75(10):712–727, July 2002.

[37] O. L. V. Costa and S. Guerra. Stationary filter for linear minimum mean square error estimator of discrete-time Markovian jump systems. *IEEE Trans. Automat. Control*, 47(8):1351–1356, August 2002.

[38] O. L. V. Costa and R. P. Marques. Mixed H_2/H_∞-control of discrete-time Markovian jump linear systems. *IEEE Trans. Automat. Control*, 43:95–100, 1998.

[39] O. L. V. Costa and R. P. Marques. Robust H_2-control for discrete-time Markovian jump linear systems. *Int. J. Contr.*, 73(1):11–21, January 2000.

[40] O. L. V. Costa and E. F. Tuesta. Finite horizon quadratic optimal control and a separation principle for Markovian jump linear systems. *IEEE Trans. Automat. Control*, 48(10):1836–1842, October 2003.

[41] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

[42] R. V. Cox. Speech coding. In *The Digital Signal Processing Handbook*, chapter 45, pages 45.1–45.19. CRC and IEEE Press, 1998.

[43] L. D. Davisson. Information rates for data compression. In *IEEE WESCON*, 1968. Session 8, Paper 1.

[44] C. E. de Souza and M. D. Fragoso. H_∞ filtering for Markovian jump linear systems. *Int. J. Sys. Sci.*, 33(11):909–915, November 2002.

[45] J. B. R. do Val and T. Basar. Receding horizon control of jump linear systems and a macroeconomic policy. *J. Economic Dynamics and Control*, 23:1099–1131, 1999.

[46] J. B. R. do Val, J. C. Geromel, and O. L. V. Costa. Uncoupled Riccati iterations for the linear quadratic control problem of discrete-time Markov jump linear systems. *IEEE Trans. Automat. Control*, 43(12):1727–1733, December 1998.

[47] A. Doucet, A. Logothetis, and V. Krishnamurthy. Stochastic sampling algorithms for state estimation of jump Markov linear systems. *IEEE Trans. Automat. Control*, 45(1):188–202, January 2000.

[48] F. Dufour and R. J. Elliott. Adaptive control of linear systems with Markov perturbations. *IEEE Trans. Automat. Control*, 43(3):351–372, March 1998.

[49] A. A. El Gamal and T. M. Cover. Achievable rates for multiple descriptions. *IEEE Trans. Inform. Theory*, IT-28(6):851–857, November 1982.

[50] N. Elia. When Bode meets Shannon: Control-oriented feedback communication systems. *IEEE Trans. Automat. Control*, 49(9):1477–1488, September 2004.

[51] P. Elias. Predictive coding–part I. *IRE Trans. Inform. Theory*, IT-1(1):16–24, March 1955.

[52] P. Elias. Predictive coding–part II. *IRE Trans. Inform. Theory*, IT-1(1):24–33, March 1955.

[53] R. J. Elliott, F. Dufour, and D. D. Sworder. Exact hybrid filters in discrete time. *IEEE Trans. Automat. Control*, 41(12):1807–1810, December 1996.

[54] European Broadcast Union. Sound quality assessment material recordings for subjective tests. Tech. 3253-E, April 1988. Available on-line at `http://sound.media.mit.edu/mpeg4/audio/sqam/`.

[55] Y. Fang. A new general sufficient condition for almost sure stability of jump linear systems. *IEEE Trans. Automat. Control*, 42(3):378–382, March 1997.

[56] N. Farvardin and J. W. Modestino. Rate–distortion performance of DPCM schemes for autoregressive sources. *IEEE Trans. Inform. Theory*, IT-31(3):402–418, May 1985.

[57] T. L. Fine. The response of a particular nonlinear system with feedback to each of two random processes. *IEEE Trans. Inform. Theory*, IT-14(2):255–264, March 1968.

[58] J. Gao, B. Huang, and Z. Wang. LMI-based robust H_∞ control of uncertain linear jump systems with time-delay. *Automatica*, 37:1141–1146, 2001.

[59] M. Gastpar, B. Rimoldi, and M. Vetterli. To code, or not to code: Lossy source–channel communication revisited. *IEEE Trans. Inform. Theory*, 49(5):1147–1158, 2003.

[60] J. C. Geromel. Optimal linear fitering under parameter uncertainty. *IEEE Trans. Signal Proc.*, 47(1):168–175, January 1999.

[61] N. L. Gerr and S. Cambanis. Analaysis of adaptive differential PCM of a stationary Gauss–Markov input. *IEEE Trans. Inform. Theory*, IT-33(3):350–359, May 1987.

[62] A. Gersho. The channel splitting problem and modulo-PCM coding. Bell Laboratories Memorandum for Record (not archived), October 1979.

[63] A. Gersho. Advances in speech and audio compression. *Proc. IEEE*, 82(6):900–918, June 1994.

[64] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Acad. Pub., Boston, MA, 1992.

[65] J. D. Gibson. Adaptive prediction in speech differential encoding systems. *Proc. IEEE*, 68(4):488–525, April 1980.

[66] H. Gish. Optimum quantization of random sequences. Technical Report AD 656 042, Defense Supply Agency, May 1967.

[67] H. Gish and J. P. Pierce. Asymptotically efficient quantizing. *IEEE Trans. Inform. Theory*, IT-14(5):676–683, September 1968.

[68] N. Gogate, D.-M. Chung, S. S. Panwar, and Y. Wang. Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding. *IEEE Trans. Circuits Syst. Video Technol.*, 12(9):777–792, September 2002.

[69] V. K Goyal. Multiple description coding: Compression meets the network. *IEEE Sig. Proc. Mag.*, 18(5):74–93, September 2001.

[70] V. K Goyal. On the side distortion lower bound in "Generalized multiple description coding with correlating transforms". Available at `http://www.rle.mit.edu/stir/`, November 2002.

[71] V. K Goyal, J. A. Kelner, and J. Kovačević. Multiple description vector quantization with a coarse lattice. *IEEE Trans. Inform. Theory*, 48(3):781–788, March 2002.

[72] V. K Goyal and J. Kovačević. Generalized multiple description coding with correlating transforms. *IEEE Trans. Inform. Theory*, 47(6):2199–2224, September 2001.

[73] V. K Goyal, J. Kovačević, and J. A. Kelner. Quantized frame expansions with erasures. *Appl. Comput. Harm. Anal.*, 10(3):203–233, May 2001.

[74] R. M. Gray. Quantization noise spectra. *IEEE Trans. Inform. Theory*, 36(6):1220–1244, November 1990.

[75] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory*, 44(6):2325–2383, October 1998.

[76] R. M. Gray and T. G. Stockham, Jr. Dithered quantizers. *IEEE Trans. Inform. Theory*, 39(3):805–812, May 1993.

[77] U. Grenander and G. Szegö. *Toeplitz Forms and Their Applications.* Univ. California Press, Berkeley, CA, 1958.

[78] O. G. Guleryuz and M. T. Orchard. On the DPCM compression of Guassian autoregressive sequences. *IEEE Trans. Inform. Theory*, 47(3):945–956, March 2001.

[79] E. Gündüzhan and K. Momtahan. A linear prediction based packet loss concealment algorithm for PCM coded speech. *IEEE Trans. Speech Audio Proc.*, 9(8):778–785, November 2001.

[80] J. Hagenauer. Source-controlled channel decoding. *IEEE Trans. Comm.*, 43(9):2449–2457, September 1995.

[81] T. Hashimoto and S. Arimoto. On the rate-distortion function of non-stationary autoregressive processes. *IEEE Trans. Inform. Theory*, 26(4):478–480, April 1980.

[82] B. A. Heng. *Adaptive Multiple Description Mode Selection for Error Resilient Video Communications*. PhD thesis, Mass. Inst. Tech., June 2005.

[83] A. Ingle and V. A. Vaishampayan. DPCM system design for diversity systems with applications to packetized speech. *IEEE Trans. Speech Audio Proc.*, 3(1):48–57, January 1995.

[84] E. Janardhanan. Differential PCM systems. *IEEE Trans. Comm.*, COM-27(1):82–93, January 1979.

[85] N. S. Jayant. Digital coding of speech waveforms: PCM, DPCM, and DM quantizers. *Proc. IEEE*, 62(5):611–632, May 1974.

[86] N. S. Jayant. Subsampling of a DPCM speech channel to provide two "self-contained" half-rate channels. *Bell Syst. Tech. J.*, 60(4):501–509, April 1981.

[87] N. S. Jayant and S. W. Christensen. Effects of packet losses in waveform coded speech and improvements due to an odd–even sample-interpolation procedure. *IEEE Trans. Comm.*, COM-29(2):101–109, February 1981.

[88] N. S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, Englewood-Cliffs, NJ, 1984.

[89] W. Jiang and A. Ortega. Multiple description speech coding for robust communication over lossy packet networks. In *Proc. IEEE Int. Conf. Multimedia & Expo*, volume 1, pages 444–447, New York, July–August 2000.

[90] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME Trans. Part D*, 82:35–45, 1960.

[91] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *ASME Trans. Part D*, 83:95–108, 1961.

[92] J. C. Kieffer. Stochastic stability for feedback quantization schemes. *IEEE Trans. Inform. Theory*, IT-28(2):248–254, March 1982.

[93] E. G. Kimme and F. F. Kuo. Synthesis of optimal filters for a feedback quantization scheme. *IEEE Trans. Circuits Syst.*, 10(3):405–413, September 1963.

[94] J. Kovačević, P. L. Dragotti, and V. K Goyal. Filter bank frame expansions with erasures. *IEEE Trans. Inform. Theory*, 48(6):1439–1450, June 2002.

[95] N. N. Krasovskii and E. A. Lidskii. Analytical design of controllers in systems with random attributes I, II, III. *Automation Remote Contr.*, 22:1021–1025, 1141–1146, 1289–1294, 1961.

[96] G. Kubin and W. B. Kleijn. Multiple-description coding (MDC) of speech with an invertible auditory model. In *Proc. IEEE Workshop Speech Coding*, pages 81–83, Porvoo, Finland, June 1999.

[97] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice-Hall, Englewood Cliffs, NJ, 1986.

[98] D. H. Lee and D. L. Neuhoff. Asymptotic distribution of the errors in scalar and vector quantizers. *IEEE Trans. Inform. Theory*, 42(2):446–460, March 1996.

[99] W. S. Lee, M. R. Pickering, M. R. Frater, and J. F. Arnold. A robust codec for transmission of very low bit-rate video over channels with bursty errors. *IEEE Trans. Circuits Syst. Video Technol.*, 10(8):1403–1412, December 2000.

[100] D. LeGall. MPEG: A video compression standard for multimedia applications. *Comm. ACM*, 34(4):46–58, April 1991.

[101] Y. Li and H. J. Chizeck. Controllability, stabilizability, and continuous-time Markovian jump linear quadratic control. *IEEE Trans. Automat. Control*, 35(7):777–788, July 1990.

[102] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy. Quantization and dither: A theoretical survey. *J. Audio Eng. Soc.*, 40(5):355–375, May 1992.

[103] X. Liu and A. Goldsmith. Kalman filtering with partial observation losses. In *Proc. IEEE Conf. Dec. & Contr.*, pages 4180–4186, December 2004.

[104] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, IT-28(2):129–137, March 1982. Originally an unpublished Bell Telephone Laboratories tech. memo., July 31, 1957.

[105] M. Luby. LT codes. In *Proc. IEEE Symp. Found. Comp. Sci.*, pages 271–280, Vancouver, November 2002.

[106] M. S. Mahmoud and P. Shi. Robust control for Markovian jump linear discrete-time systems with unknown nonlinearities. *IEEE Trans. Circuits Syst.–I: Fundamental Theory and Appl.*, 49(4):538–542, April 2002.

[107] A. Majumdar and K. Ramchandran. PRISM: an error-resilient video coding paradigm for wireless networks. In *Proc. First Int. Conf. on Broadband Networks*, pages 478–485, 2004.

[108] S. Makhoul. Linear prediction: A tutorial overview. *Proc. IEEE*, 63(4):561–580, April 1975.

[109] M. Mariton and P. Bertrand. Robust jump linear quadratic control: A mode stabilizing solution. *IEEE Trans. Automat. Control*, AC-30(11):1145–1147, November 1985.

[110] J. Max. Quantizing for minimum distortion. *IRE Trans. Inform. Theory*, IT-6(1):7–12, March 1960.

[111] D. Q. Mayne. A solution of the smoothing problem for linear dynamic systems. *Automatica*, 4:73–92, 1966.

[112] S. E. Miller. New transmission configuration. Bell Laboratories lab notebook #55637, May 1978.

[113] S. E. Miller. Fail-safe transmission without standby facilities. Technical Report TM80-136-2, Bell Laboratories, August 1980.

[114] N. Nahi. Optimal recursive estimation with uncertain observation. *IEEE Trans. Inform. Theory*, IT-15(4):944–948, July 1969.

[115] G. N. Nair and R. J. Evans. Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM J. Contr. Optimiz.*, 43(2):413–436, 2004.

[116] M. Naraghi-Pour and D. L. Neuhoff. Mismatched DPCM encoding of autoregressive processes. *IEEE Trans. Inform. Theory*, 36(2):296–304, March 1990.

[117] D. L. Neuhoff. On the asymptotic distribution of errors in vector quantization. *IEEE Trans. Inform. Theory*, 42(2):461–468, March 1996.

[118] J. B. O'Neal, Jr. Signal-to-quantizing-noise ratios for differential PCM. *IEEE Trans. Commun. Syst.*, 19:568–570, August 1971.

[119] J. B. O'Neal, Jr. Differential pulse-code modulation (PCM) with entropy coding. *IEEE Trans. Inform. Theory*, IT-22(2):169–174, March 1976.

[120] L. Ozarow. On a source-coding problem with two channels and three receivers. *Bell Syst. Tech. J.*, 59(10):1909–1921, December 1980.

[121] Z. Pan and T. Basar. Zero-sum differential games with random structures and applications in H_∞ control of jump linear systems. In *Int. Symp. Dynamic Games and Applications*, pages 466–480, Quebec, Canada, 1994 1994.

[122] S. S. Pradhan, R. Puri, and K. Ramchandran. n-channel symmetric multiple descriptions—Part I: (n, k) sourc–channel erasure codes. *IEEE Trans. Inform. Theory*, 50(1):47–61, January 2004.

[123] R. Puri, S. S. Pradhan, and K. Ramchandran. n-channel symmetric multiple descriptions—Part II: An achievable rate–distortion region. *IEEE Trans. Inform. Theory*, 51(4):1377–1392, April 2005.

[124] R. Puri and K. Ramchandran. PRISM: a new robust video coding architecture based on distributed compression principles. In *Proc. 40th Allerton Conf. on Communication, Control, and Computing*, Allerton, IL, October 2002.

[125] M. Quirk. Diversity coding for communication systems. Bell Laboratories Engineer's Notes (not archived), December 1979.

[126] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, second edition, 2002.

[127] D. O. Reudink. The channel splitting problem with interpolative coders. Technical Report TM80-134-1, Bell Laboratories, October 1980.

[128] I. E. G. Richardson. *H.264 and MPEG-4 Video Compression*. John Wiley & Sons, New York, 2003.

[129] A. Sahai and S. Mitter. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link, parts I and II. *IEEE Trans. Inform. Theory*, November 2004. Submitted. Preprint available at `http://www.eecs.berkeley.edu/~sahai/Papers/control-part-I.pdf` and `control-part-II.pdf`.

[130] A. Sahai and S. Mitter. Source coding and channel requirements for unstable random processes. *IEEE Trans. Inform. Theory*, 2005. Submitted. Preprint available at `http://www.eecs.berkeley.edu/~sahai/Papers/anytime.pdf`.

[131] K. Sayood and J. C. Borkenhagen. Use of residual redundancy in the design of joint source/channel coders. *IEEE Trans. Comm.*, 39(6):838–846, June 1991.

[132] G. Schuller and A. Härmä. Low delay audio compression using predictive coding. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume II, pages 1853–1856, Orlando, FL, May 2002.

[133] G. Schuller, J. Kovačević, F. Masson, and V. K. Goyal. Robust low-delay audio coding using multiple descriptions. *IEEE Trans. Speech Audio Proc.*, 13(5):1014–1024, September 2005.

[134] S. D. Servetto and K. Nahrstedt. Broadcast-quality video over IP. *IEEE Trans. Multimedia*, 3(1):162–173, March 2001.

[135] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, July 1948. Continued 27:623–656, October 1948.

[136] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Int. Conv. Rec., part 4*, 7:142–163, 1959. Reprinted with changes in *Information and Decision Processes*, ed. R. E. Machol, McGraw-Hill, New York, 1960, pp. 93–126.

[137] P. Shi and E. K. Boukas. H_∞ control for Markovian jumping linear systems with parametric uncertainty. *J. Optimiz. Theory and Appl.*, 95:75–99, 1997.

[138] R. Singh and A. Ortega. Consistent estimation of erased data in a DPCM based multiple description coding system. *J. VLSI Sig. Proc.*, 34(1–2):9–28, May 2003.

[139] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. In *Proc. IEEE Conf. Dec. & Contr.*, volume 1, pages 701–708, Maui, Hawaii, December 2003.

[140] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Trans. Automat. Control*, 49(9):1453–1464, September 2004.

[141] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry. Distributed control applications within sensor networks. *Proc. IEEE*, 91(8):1235–1246, August 2003.

[142] W. R. Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, Reading, MA, 1994.

[143] P. Stoica and I. Yaesh. Jump Markovian-based control of wing deployment for an uncrewed aircraft. *J. Guidance*, 25:407–411, 2002.

[144] D. D. Sworder. Feedback control of a class of linear systems with jump parameters. *IEEE Trans. Automat. Control*, AC-14:60–63, 1969.

[145] S. Tatikonda and S. Mitter. Control over noisy channels. *IEEE Trans. Automat. Control*, 49(7):1196–1201, July 2004.

[146] S. Tatikonda, A. Sahai, and S. Mitter. Stochastic linear control over a communication channel. *IEEE Trans. Automat. Control*, 49(9):1549–1561, September 2004.

[147] V. Vaishampayan and A. A. Siddiqui. Speech predictor design for diversity communication systems. In *Proc. IEEE Workshop on Speech Coding for Telecommunications*, pages 69–70, Annapolis, MD, September 1995.

[148] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Trans. Inform. Theory*, 39(3):821–834, May 1993.

[149] V. A. Vaishampayan. Application of multiple description codes to image and video transmission over lossy networks. In *Proc. Int. Workshop on Packet Video*, March 1996.

[150] V. A. Vaishampayan and J.-C. Batllo. Asymptotic analysis of multiple description quantizers. *IEEE Trans. Inform. Theory*, 44(1):278–284, January 1998.

[151] V. A. Vaishampayan and J. Domaszewicz. Design of entropy-constrained multiple-description scalar quantizers. *IEEE Trans. Inform. Theory*, 40(1):245–250, January 1994.

[152] V. A. Vaishampayan and S. John. Balanced interframe multiple description video compression. In *Proc. IEEE Int. Conf. Image Proc.*, volume 3, pages 812–816, Kobe, Japan, October 1999.

[153] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto. Multiple-description vector quantization with lattice codebooks: Design and analysis. *IEEE Trans. Inform. Theory*, 47(5):1718–1734, July 2001.

[154] R. Venkataramani, G. Kramer, and V. K Goyal. Multiple description coding with many channels. *IEEE Trans. Inform. Theory*, 49(9):2106–2114, September 2003.

[155] Y. Wang, M. T. Orchard, V. Vaishampayan, and A. R. Reibman. Multiple description coding using pairwise correlating transforms. *IEEE Trans. Image Proc.*, 10(3):351–366, March 2001.

[156] H. S. Witsenhausen. On source networks with minimal breakdown degradation. *Bell Syst. Tech. J.*, 59(6):1083–1087, July–August 1980.

[157] H. S. Witsenhausen. Minimizing the worst-case distortion in channel splitting. *Bell Syst. Tech. J.*, 60(8):1979–1983, October 1981.

[158] H. S. Witsenhausen and A. D. Wyner. Source coding for multiple descriptions II: A binary source. *Bell Syst. Tech. J.*, 60(10):2281–2292, December 1981.

[159] J. K. Wolf, A. D. Wyner, and J. Ziv. Source coding for multiple descriptions. *Bell Syst. Tech. J.*, 59(8):1417–1426, October 1980.

[160] W. M. Wonham. On a matrix Riccati equation of stochastic control. *SIAM J. Contr.*, 6(4):681–697, 1968.

[161] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inform. Theory*, IT-22(1):1–10, January 1976.

[162] R. Zamir. Gaussian codes and Shannon bounds for multiple descriptions. *IEEE Trans. Inform. Theory*, 45(6):2629–2635, November 1999.

[163] Z. Zhang and T. Berger. Multiple description source coding with no excess marginal rate. *IEEE Trans. Inform. Theory*, 41(2):349–357, March 1995.

[164] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ, 1995.