Distributed Control and Computing: Optimal Estimation, Error-Correcting Codes, and Interactive Protocols

Thesis by

Ravi Teja Sukhavasi

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2012

(Defended May 09, 2012)

To

my parents

Acknowledgements

Reflecting upon the years of graduate school that culminated in this thesis, it gives me great pleasure to thank the many people who made it possible.

I owe my deepest gratitude to my adviser, Prof. Babak Hassibi. Without his vision, guidance and kindness, this thesis would have been but a dream. I have benefited immensely from Babak's deep understanding of the subject and his willingness to spend hours on end sharing his insights and ideas. His exceptional teaching qualities, his remarkable ability to identify the essence of a problem and his broad scholarship will always be an inspiration to me.

I would like to thank Prof. Palghat P. Vaidyanathan, Prof. Tracey Ho, Prof. Richard Murray and Prof. John Doyle for serving on my Ph.D. examination committee, for insightful comments on my thesis manuscript and for providing valuable advice throughout my course of study.

I would like to acknowledge the support of my research in part by the National Science Foundation, The Air Force Office for Scientific Research, the Lee Center for Advanced Networking and most importantly the American taxpayer.

This thesis benefited from academic interactions with many people. In particular, I am grateful to Niranjan Balachandran and Shirin Jalali for many useful technical discussions and to my undergradudate mentees Noele Norris, Yash Deshpande, Marland Sitt, and Yishun Dong whose research helped improve my own understanding of the subject.

I am very thankful to our group secretary, Shirley Beatty, for her professional assistance on administrative matters and her kindness and care. I will always be grateful to all my colleagues for making my work life pleasant and exciting. I will miss

the the countless hours spent with my officemates, especially with Wei Mao, Matthew Thill, and Elizabeth Bodine-Baron either chatting about pretty much everything under the sun or eating spicy Hunan food or religiously watching Hollywood summer blockbusters.

I will walk away from Caltech with a treaure trove of great experiences and long lasting friendships. I would like to thank Jayakrishnan Nair, Piya Pal, Vikas Trivedi, Srivatsan Hulikal, Niranjan Srinivas, Pinkesh Patel, Bharat Penmacha, Sowmya Chandrasekhar, Vijay Natraj, Kaushik Sengupta, and many more friends whose company and kindness were key to my happiness and success during my graduate student years. I would especially like to thank Sushree Mishra for her priceless friendship and support. I am deeply grateful to the Organization of the Associated Students of the Indian Subcontinent (OASIS) and the International Student Programs (ISP) office for their constant support, the many wonderful people that I met through them and for helping create a vibrant atmosphere for international students like me.

With its beautiful campus, world class facilities, friendly staff and wonderful people, Caltech has provided me with an enriching environment to grow not only as a researcher but also as a human being, and for this I cannot be more grateful. I have also thoroughly enjoyed attending the numerous campus events throughout the year, such as the olive harvest festival, π day, and countless inspiring public lectures. I would also like to thank Ernie's food truck, where I got most of my lunches, for the delicious and inexpensive food.

Lastly and most importantly, I would like to thank my parents, Anjani Sukhavasi and Brahmanandam Sukhavasi, for their unconditional love, support and all the hardships that they have gone through to provide me with the best opportunities to prosper. I am honored to have such wonderful parents, and I dedicate this thesis to them as an inadequate but sincere expression of gratitude and love.

Abstract

Emerging applications of networked control and distributed computing are characterized by decentralization of information and the need to exchange it over potentially unreliable communication networks. This results in novel interactive communication scenarios that are incompatible with conventional information and coding theoretic approaches. To address this gap, through the early and late 1990's, a new information theoretic notion called *anytime reliability* and a new coding paradigm called *tree codes* were proposed. Although the central role of tree codes in several interactive communication problems such as distributed control and computing has been well understood, there have been no practical constructions till date. For the first time, we have an explicit ensemble of linear tree codes with efficient encoding and decoding for the class of erasure channels. In the process, we have developed novel non-asymptotic sufficient conditions on the kind of communication reliability required to stabilize control systems over noisy channels. We also study the application of tree codes to interactive protocols over erasure networks and illustrate their benefits through the example of average consensus.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fueled by rapid advances in embedded systems technology and communications infrastructure, cheaply available *smart* devices with small form factors, capable of sensing, computing and wireless communications, have proliferated throughout many applications. These advances have enabled monitoring and data collection from an unprecedented variety of areas encompassing weather and environment, medical care, energy consumption, vehicular traffic, public spaces, structural health monitoring of man-made constructions and even online social networks.

The next logical step in this evolution is to use this data to control and influence the physical world in an automated manner with minimal human intervention. Possible instances of this new paradigm include the smart grid that is capable of meeting fluctuating demands by automatically augmenting or switching between various renewable/nonrenewable power supplies [2], intelligent highway systems [42], smart homes that automatically adjust according to the needs of the occupants [43], networked city services and formation flying of underwater/aerial vehicles/satellites [65], to mention just a few. Widely referred to as *cyberphysical systems* and/or *networked control systems*, they have inspired a lot of research and developmental activity of late.

Essential to understaning and realizing such networked control systems in practice is a convergence of tools from computing, communications and control. There has been significant effort by the research community in this direction in recent years [1, 44, 57, 70, 84]. Two important features of networked control systems are

decentralization of information and the need to exchange it over potentially unreliable communication networks. Consequently, one of the key challenges (e.g., [70]) in building future networked control systems is to integrate information theory and control theory, two fields that have traditionally developed almost completely independently of each other. The work presented in this thesis is motivated by this challenge and is broadly made up of two parts, namely decentralized estimation and communication for control.

1.1 Control Theory

Control theory, at its simplest, is concerned with regulating the behavior of dynamical systems through output feedback. A typical control system is comprised of the plant or the dynamical system, the measurement unit which measures the output of the plant, the control unit or controller which uses the output to determine appropriate feedback and the actuation unit which applies the feedback determined by the controller. This is illustrated in Figure 1.1.

Feedback control played a key role in the development of technologies ranging from power, transportation, and manufacturing to communication, and data storage/retrieval. For example, man's journey to the moon would not have been possible without feedback control. One of the earliest applications of feedback control is the centrifugal governor which is a primitive cruise control used in early Watt steam engines. More routine applications include autopilots in aviation, regulation and control of the electrical power grid, and high-accuracy positioning of read/write heads in disk drives. In most traditional applications, the associated control systems are fully centralized, i.e., the plant, measurement, control and actuation units are all hardwired together. A very rich theory of classical feedback control has been developed over the past century, key milestones include Nyquist stability criterion [72], Wiener filter [108], state-space approach and Kalman filteirng [52, 53], H^∞-control [26, 40], LQG control and the separation principle [50]. Common to all these developments is the traditional model of control systems depicted in Figure 1.1.

In contrast, cyber-physical systems are characterized by different levels of decentralization in their structure. At a high level, the measurement unit and the control unit are not colocated. In addition, each is comprised of arrays of sensors and actuators that in turn communicate with each other over a network as depicted in Figure 1.2. We already see instances of this in the form of control area networks (CAN) in modern vehicles where different control sub-systems such as window controllers, cruise control system and headlight positioning systems communicate over a shared network layer. As a result of this decentralization, most of the classical control techniques do not apply directly.

Early work on decentralized control appeared in [109] where Witsenhausen used a simple instance of a decentralized optimal control problem to disprove the conjecture that affine control laws are optimal for sufficiently centralized linear-quadratic-Gaussian (LQG) control systems. There has subsequently been a lot of work on studying the effect of nonclassical information structures that arise in decentralized control of which [109] is an example. An early survey can be found in [88] and more recent papers include [8, 77, 83]. This body of work assumes that communication between different components is instantaneous and perfect.

Research on the effects of communication constraints in distributed control did not appear until more recent years [9, 15, 110, 111]. The presence of communication channels in the feedback loop of control systems raises important fundamental questions on conventional information theoretic approaches for achieving communication reliability. Control theory and information theory make incompatible sets of modeling assumptions on real life systems. Whereas control theorists tend to assume that all communication is reliable and delay free, information theorists can guarantee reliability only in the asymptotic limit of large delay. Resolving this dichotomy between delay and reliability is key to developing a more integrated systems theory of networked control systems. We elaborate on this below.

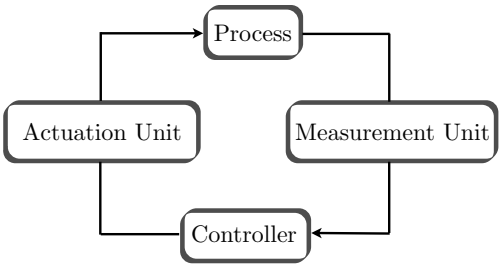


Figure 1.1: A control system

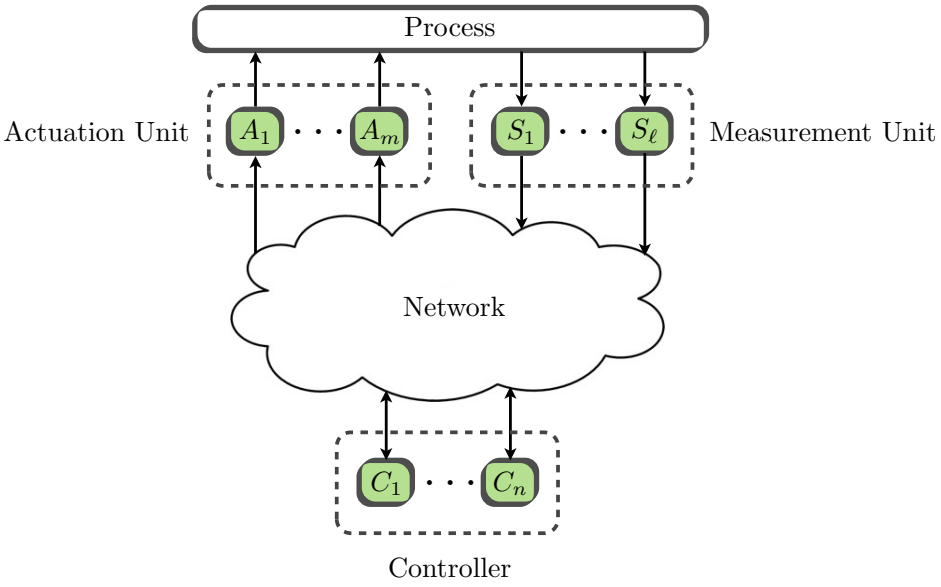


Figure 1.2: A networked control system: S_i, A_i and C_i denote sensors, actuators and controllers respectively

1.2 Information Theory and Control Theory

In his landmark paper [94], Shannon laid the mathematical foundations of modern communication systems. Influenced by Turing's work, Shannon emphasized treating messages as discrete symbols as opposed to continuous waveforms and coined the currency of bits. He realized that sending just one bit over a channel and reproducing it accurately at the other end is impossible but reproducing a whole bunch of them is not. This is because a channel is unpredictable over one use but becomes very predictable over several uses, thanks to law of large numbers. For example, if the channel flips each bit with probability p and independently of the rest, then over n channel uses it would flip approximately np bits with a very high probability. This motivated the idea of block coding.

When encoding a message, we break it up into blocks of k bits each, add redundancy and encode each block independently into a larger block of n bits (e.g., Figure 1.3). The rate of the code is k/n and the optimal decoder selects the most likely input given the channel outputs. Shannon showed that the corresponding probability of decoding error vanishes to zero if and only if the rate is smaller than the channel capacity. This is a beautiful theory and an elegant result. Though Shannon only showed existence of block codes that achieve capacity, thanks to sixty years of coding theory, we now have several practical codes that reach the Shannon limits in many ways. Some examples include low density parity check (LDPC) codes with message passing decoding [32, 61, 81, 82], convolutional codes with Viterbi decoding [105], algebraic geometric codes with Berlekemp-Massey or list decoding [13, 39, 63], Polar codes [5] and so on.

All these techniques achieve reliability at the expense of encoding and decoding delay. The greater the delay, higher the reliability. Delay is seen as a necessary evil and over time has received much less attention. This was not a problem in traditional communication systems since most applications were delay tolerant, e.g., cellular speech communication where encoding/decoding delay is imperceptible. But such conventional coding techniques are not appropriate in the context of networked

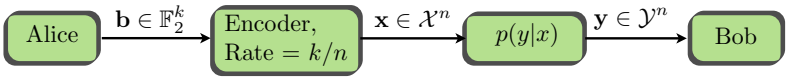


Figure 1.3: Block coding

control systems since delay can severely degrade the performance of the system and even result in instability. This marks a key difference between the philosophy of information theory and that of control theory. Control theory deals with real time constraints where as information theory largely lives in asymptopia. Furthermore, information theory largely deals with one way communication while in distributed control, communication is interactive, i.e., the plant measurements to be encoded are determined by the control inputs which in turn are determined by how the controller decodes the corrupted plant measurements.

To address these incompatibilities, through the early and late 1990s, a new information theoretic notion called *anytime reliability* and a new coding paradigm called *tree codes* were proposed in [84] and independently in [92] respectively. Tree codes are central to several distributed applications including distributed computation and distributed control. But there were no explicit constructions of tree codes since and the subject has remained in the realm of pure theory. For the first time, we have an explicit ensemble of tree codes with efficient encoding and decoding for a class of communications channels called erasure channels which are used in practice to model links under packetized communication, this includes the internet and wireless links. We also study their application to implementing protocols within a group of agents connected by a communication graph with erasure links. We will explain the key contributions of this thesis in greater detail in the following section.

1.3 Contributions

In this section, we briefly review the contents of each chapter and outline the main contributions. The chapters can be read mostly independently from each other. This is particularly true of Chapter 2 which focuses mainly on decentralized estimation

while the remaining chapters focus on error-correcting codes for distributed control and general interactive communication problems.

1.3.1 Decentralized Estimation

Rapid advances in embedded systems technology have lead to a proliferation of cheap and often low quality sensors with wireless communication capabilities for measuring and recording various physical phenomenon. A wireless sensor network refers to a network of such sensors used to monitor large physical spaces. Initially motivated by military applications like battlefield surveillance, today they are widely used in many areas ranging from industrial applications such as process monitoring and control, and structural health monitoring, environmental applications such as endangered species monitoring, and consumer applications like automatic climate control in homes and offices. The most common sensor network architecture involves collecting the sensor measurements over a network at a fusion center which aggregates the measurements and uses them to perform a desired task. There is often a scheduling algorithm that schedules different subsets of sensor to make measurements at each time which are then quantized and communicated back to the fusion center. We study the problem of optimal estimation using quantized measurements with a focus on sensor network applications in Chapter 2.

There are a number of applications that have natural power and bandwidth constraints for reasons ranging from stealth, desired longevity (e.g., due to difficulty in replacing the sensors) and shared communication medium with other technologies, etc. Consequently, the measurements are often coarsely quantized and hence quantization effects cannot be ignored. In contrast to classical LQG cotrol where a single measurement unit has access to all the analog measurements, there is no single entity in the network that has access to all the analog sensor measurements. As a result, classical estimation techniques do not apply in this context.

We considered the problem of optimally estimating the state of the system using quantized sensor measurements in the case where the plant is described by a linear

Gaussian State-space model and the measurements are linear. Due to the non-linear nature of the problem, analytical approches fail in this setup and a natural alternative often suggested in the literature is to use particle filters. Particle filtering is a numerical technique which is best described as a sequential monte carlo algorithm. We exploit the State-space structure of the plant to propogate most of the state information analytically and use the particle filter to propogate only the essential nonlinearity in the estimation algorithm. The result is an extremely efficient numerical filter that can optimally track the state using far fewer particles (up to two orders of magnitude fewer) than conventional particle filters. We call this the Kalman-Like Particle Filter (KLPF) and describe it in detail in Chapter 2. We also present new results on the distribution of the state conditioned on quantized measurements and conclude the Chapter with simulations that compare the performance of the KLPF with those from the literature.

1.3.2 Distributed Control

While in Chapter 2, we focus primarily on the effects of quantization on optimal estimation, in the remaining chapters we shift focus to the case where communication is not only rate limited but also stochastic and noisy. A natural approach in this case would be to quantize and packetize each plant measurement and communicate it to the controller. Motivated by such a setup, the authors in [95] considered the problem of optimal LQG control when plant measurements are subject to erasures. It was shown that if the plant is open-loop unstable, closed-loop mean-squared stability is not possible whenever the erasure rate exceeds a threshold that is determined by the plant dynamics. This suggests that conventional notions of communication reliability centered around block coding are inadequate when considering communication channels that are in the feedback loop of control systems.

The problem of stabilizing an unstable system over a noisy channel captures the essential complexity of coding for control. An information theoretic framework for this problem was first studied by Sahai in [84] where the notion of anytime reliability

was proposed as the right metric for measuring communication reliability for control. Roughly speaking, an encoder and decoder pair over a communication channel is said to be $(R, \beta)-$anytime reliable if the communication rate is R and the probability of incorrectly decoding a bit that was sent d time steps ago decays exponentially in d with exponent β. [84] developed sufficient conditions on the rate R and exponent β for stabilizing scalar unstable processes in closed-loop which are also necessary if there is perfect channel feedback from the decoder to the encoder. In particular, if the plant eigen value is λ, then one needs $R > \log|\lambda|$ and $\beta > 2\log|\lambda|$ for mean-squared stability. These bounds were extended to the case of vector-valued processes with channel feedback in [87].

The plant is said to be mean-squared stable in closed-loop if the second moment of the state is asymptotically finite. The sufficient conditions on the rate and exponent proposed in [84, 87] that ensure closed loop stability are asymptotic in nature. The same is true of the sufficient conditions presented in [62,64,66,71,100] which deal with the case where the communication channels are rate limited but noiseless. In other words, the second moment of the state will be finite but can be arbitrarily large. In practice, one cares about keeping the state small rather than just finite. Motivated by this spirit, we present in Chapter 3 novel non-asymptotic sufficient conditions on the rate and exponent for closed-loop stability of linear State-space processes over noisy channels. Moreover we consider the case where there is no feedback from the decoder to the encoder. To the best of our knowledge, this has not been considered before in the literature. Even though the sufficient conditions developed in Chapter 3 are non-asymptotic, the thresholds on the rate and exponent depend only on the system parameters, in particular, on the co-efficients of the characteristic polynomial of the plant. We also show that the thresholds are asymptotically tight. In the process of proving these sufficient conditions, we developed novel filtering algorithms for tracking the state using quantized measurements.

1.3.3 Error-Correcting Codes for Control

The sufficient conditions on the rate and exponent of anytime reliable codes developed in [86, 87] and Chapter 3 are predicated upon the existence of error-correcting codes that achieve such reliabilities. One needs so called tree codes in order to achieve anytime reliability over memoryless channels under maximum-likelihood (ML) decoding. Tree codes first appeared in the work of Schulman [79,92] in the context of distributed computation. Schulman used tree codes to simmulate interactive protocols between a network of agents and showed that tree codes exist. The main contribution of [92] is to effectively provide an interactive analogue of Shannon's noisy channel coding theorem which considered one way communication. In particular, [92] proved that one can simulate any interactive protocol between a pair of agents while suffering a constant slowdown and guaranteeing an error probability that is exponentially small in the length of the protocol. The focus was on achieving constant slowdown no matter how small the constant was. Distributed control can also be viewed as an instance of interactive communication but the emphasis is much more on the constants. In this case, the rate R which is the slowdown and the exponent β corresponding to the error probability need to be simultaneously large enough.

Even though the significance of tree codes in interactive communication problems has been understood for nearly two decades, there have been no practical constructions till date. The existence of tree codes proved in [92] is not with high probability. This is in contrast with Shannon's results in [94] where he proved not only that capacity achieving codes exist but that almost all random codes achieve capacity. We bridge this gap in our understanding of tree codes in Chapter 4. For the first time, we showed the existence of linear tree codes with high probability. In other words, we prove that codes drawn from an appropriate ensemble of causal linear codes which we call the *Toeplitz ensemble* are $(R, \beta)-$anytime reliable with high probability for rates upto Shannon capacity and exponent up to the expurgated exponent [11]. This significantly improves upon the known rate and exponent pairs for which anytime reliable codes are known to exist.

Tree codes require ML decoding at each time step to be anytime reliable. Performing ML decoding at each time step is impractical for most communication channels, e.g., for the binary symmetric channel the complexity of performing ML decoding at time t is $2^{\Omega(t)}$. A sequential implementation of ML decoding is presented in [92] for which the computational complexity at any decoding instant is stochastic and the probability of performing L operations decays as $L^{-\gamma}$ for some $\gamma > 0$. For small enough rates, one can show that $\gamma > 1$ and hence the average decoding complexity at any time instant is bounded. Similar observations were made in [76]. These constitute the best known results on the existence and decoding complexity of tree codes for any channel.

Note that ML decoding of linear codes over the erasure channel boils down to solving systems of linear equations. In Chapter 5, we exploit the linearity of the codes developed in Chapter 4 to propose an efficient decoding algorithm for the erasure channel. The decoding algorithm has bounded average complexity at any decoding instant for all rates up to the Shannon capacity and the probability of performing L operations decays as $2^{-\Omega\left(\sqrt[3]{L}\right)}$. This is a significant improvement over those available in the literature and works very well in practice. In Chapter 5, we also discuss possible approaches to construct efficient tree codes for the binary symmetric channel. We conclude the chapter with simulations that combine the results from chapters 3, 4, and 5.

1.3.4 Application to Interactive Protocols

In [79], the authors show that tree codes can be used to simulate protocols over a group of agents connected to each other through an arbitrary directed communication graph with noisy links. They showed that one can simulate protocols with a constant slowdown and a probability of error that vanishes exponentially fast in the length of the protocol. The results were presented for the case where the noisy channels were binary symmetric channels. We leverage the efficient tree code constructions developed in chapters 4 and 5 to develop novel algorithms to simulate protocols over

erasure networks in Chapter 6. We use the thresholds on rate and exponent of anytime reliable codes developed in Chapter 4 to provide much tighter bounds on the speed of the simulation. We apply these results to consensus problems in Chapter 7.

In a network of agents, consensus refers to the process of achieving agreement between the agents in a distributed manner. In the context of consensus problems, the unreliability of communication links between nodes has been traditionally modeled by allowing the underlying graph to vary with time. In other words, depending on the realization of the link erasures, the underlying graph at each time instant is assumed to be a subgraph of the original graph. Implicit in this model is the assumption that the erasures are symmetric: if at time t the packet from node i to node j is dropped, the same is true for the packet transmitted from node j to node i. However, in practical wireless communication systems this assumption is unreasonable and, due to the lack of symmetry, standard averaging protocols cannot guarantee that the network will reach consensus to the true average. In Chapter 7, we use coding to overcome this limitation and in general improve the performance of consensus algorithms.

Chapter 2

Kalman-Like Particle Filter

2.1 Introduction

In classical control and state estimation theory, the observer and the controller are assumed to be colocated. For partially observed Gaussian state-space models, it is well known that the minimum mean-squared error estimate of the state can be computed recursively and efficiently using the Kalman filter. With rapid advances in communication and sensing technology, there are increasingly many applications such as distributed tracking and control where measurement and control signals are communicated over noisy channels with a finite capacity. As a result, analog measurements need to be quantized before being communicated. In recent years, motivated primarily by power and bandwidth limitations in wireless sensor network applications (e.g., long endurance sensor networks for endangered species monitoring [14]), there has been a renewed interest in developing estimation algorithms using only coarsely quantized measurements. There has been a considerable amount of research in developing energy efficient algorithms for network coverage and decentralized detection and estimation using quantized sensor observations [54, 55, 59].

The problem of estimation with quantized measurements is almost as old as the Kalman filter itself. An early survey on the subject can be found in [24]. Most of the earlier techniques for estimation using quantized measurements centered on using numerical integration methods to approximate the optimal state estimate. The advent of particle filtering [6, 23, 37] created a whole new set of tools to handle non

linear estimation problems. For example, [54] proposes a particle filtering solution for optimal filtering using quantized sensor measurements. But, quantizing sensor measurements can lead to large quantization noises when the observed values are large which then leads to poor estimation accuracy. A natural alternative is to quantize the prediction error. In [110], this coding technique is referred to as the 'generalized mean coder-estimator' technique and under a very restrictive state-space model, this estimator is shown to be open-loop mean-squared stable if the quantizer rate is sufficiently high. The same scheme is independently proposed in [80, 113], where it is referred to as the 'sign of innovation' method. Under a simplifying assumption that the prior conditional state density is approximately Gaussian, the optimal filter takes a simple analytical form, which we refer to as the multiple-level-quantized Kalman filter (MLQ-KF), whose error covariance satisfies a modified Riccati recursion (MLQ-Riccati) of the type that appears in a different context in [95]. When the state is available at the sensor, [116] studies an adaptive quantization technique and proves that it can track an unstable process in open-loop with a finite mean-squared error.

If the Gaussian assumption of [80, 113] were realistic, convergence of the MLQ-Riccati must mean the convergence of the error of the MLQ-KF. [99] provides examples for which the actual error performance of MLQ-KF does not converge to the MLQ-Riccati which means that the assumption of Gaussianity is not generally true. Therefore, we present a closer examination of the conditional state density in this chapter. We derive a novel stochastic characterization of the conditional state density (see Theorem 2.1). A careful literature review reveals that related observations have been made in [25] and [3]. In particular, with some effort, [25] can be used to derive Theorem 2.1 while [3] constitutes a special case of the results presented here. Using Theorem 2.1, it is straightforward to see that the conditional state density is not Gaussian. This is to be expected given the non linear nature of quantization. In fact, it is what we refer to as a generalized closed skew normal (GCSN) distribution, which is very similar to those studied in [4, 7, 35, 36, 60, 75]. Specializing this result to state-space models, we develop a novel particle filtering approach, which we call the Kalman-like particle filter (KLPF), to estimate the state using quantized measure-

Table 2.1: Notation for Chapter 2

$u_{i:j}$	$\{u_i, \ldots, u_j\}$
$\langle X, Y \rangle$	$E(X - EX)(Y - EY)^T$
$\|X\|^2$	$\langle X, X \rangle$
$\mathcal{L}(X_1, \ldots, X_n)$	Linear span of the random variables $(X_1, \ldots, X_n)$
$\mathcal{B}(\mathbb{R})$	The Borel σ-field over the reals
$N_d(\mu, \Sigma)$	d-dim Gaussian random variable with mean μ and covariance Σ
$\phi_d(x; \mu, \Sigma)$	$\frac{1}{(2\pi)^{d/2}\sqrt{det(\Sigma)}} \exp\left(-\frac{x^T \Sigma^{-1} x}{2}\right)$
	i.e., probability density function of $N_d(\mu, \Sigma)$
$\Phi(x)$	$P(X \leq x)$, where $X \sim N(0, 1)$
$\Phi(x, ; \mu, \sigma^2)$	$P(X \leq x)$, where $X \sim N(\mu, \sigma^2)$
$\Phi(\mathcal{S}; \mu, \Sigma)$	$P(X \in \mathcal{S})$, where $X \sim N(\mu, \Sigma)$ and $\mathcal{S} \in \mathcal{B}(\mathbb{R})$

ments/innovations and study its asymptotic behavior. Finally, we show that under the information pattern studied, the classical separation property between estimation and control holds for the finite horizon LQG problem. The separation principle has been observed in several settings (see, e.g., [101,117]). It should be noted that for such separation results to be useful in practice, one needs a way to compute the MMSE estimate of the hidden state and this is primarily what we address through this work. The proposed filter requires far fewer particles than that of a particle filter applied directly to the original problem [99], as will be shown through various simulations. A preliminary version of this work appeared in [98].

The notation to be used in the rest of the chapter is summarized in Table 2.1. Also, the notion of optimality to be used throughout is mean squared error optimality.

2.2 Problem Setup and Motivation

The broader problem that one would like to solve can be cast as causal estimation of a random process $\{x_n\}$ using a quantized version, $\{q_n\}$, of the associated measurement process $\{y_n\}$. The encoding/quantization of $\{y_n\}$ into $\{q_n\}$ is determined by the information available at the encoder/observer at each time. We will limit our

attention to Gaussian state-space models, i.e., we consider the following system

$$x_{n+1} = F_n x_n + G_1 w_n + G_2 u_n \tag{2.1a}$$

$$y_n = H_n x_n + v_n \tag{2.1b}$$

where $x_n \in \mathbb{R}^d$ is the state, $y_n \in \mathbb{R}$ is the observation, and $w_n \in \mathbb{R}^p$ and $v_n \in \mathbb{R}$ are uncorrelated Gaussian white noises with zero means and covariances W and R, respectively. The initial state, x_0, of the system, is also a zero mean Gaussian with covariance P_0 and is uncorrelated with both w_n and v_n. u_n is the control input, which is set to 0 whenever we consider open-loop estimation. For a given sequence of control inputs $\{u_n\}$, the minimum mean-squared error estimate of x_n given $y_{0:n}$, which we denote with $\hat{x}^{kf}_{n|n}$, can be computed recursively using the following Kalman filtering equations (e.g., [50])

$$\hat{x}^{kf}_{n+1|n+1} = \hat{x}^{kf}_{n+1|n} + F_n P^{kf}_{n|n-1} H_n^T \left(H_n P^{kf}_{n|n-1} H_n^T + R \right)^{-1} \left(y_n - H_n \hat{x}^{kf}_{n+1|n} \right) \tag{2.2a}$$

$$\hat{x}^{kf}_{n+1|n} = F_n \hat{x}^{kf}_{n|n} + G_2 u_n, \;\; \hat{x}^{kf}_{0|-1} = 0 \tag{2.2b}$$

$$P^{kf}_{n+1|n} = F_n P^{kf}_{n|n-1} F_n^T + W - F_n P^{kf}_{n|n-1} H_n^T \left(H_n P^{kf}_{n|n-1} H_n^T + R \right)^{-1} H_n P^{kf}_{n|n-1} F_n^T \tag{2.2c}$$

and $P^{kf}_{0|-1} = P_0$.

2.2.1 Motivation

In classical LQG control, the controller is colocated with the observer and hence, at each time n, has access to $y_{0:n}$, i.e., all uncoded measurements up to time n. The controller's goal is then to determine the optimal control law u_n to minimize a given quadratic cost function. This problem is well understood. Increasingly many modern control systems employ multiple sensors and actuators that are not colocated. Towards addressing this paradigm, there has been considerable amount of work on estimation and control under communication constraints, a representative sample

being [15, 66, 71, 86, 100, 114]. Here, the observer and the controller are separated by a communication channel. Hence the observer causally quantizes the measurements $y_{0:n}$ to obtain q_n which is suitably encoded and communicated over the channel at time n.

Sensor networks provide a slightly different setting. A salient feature of [15, 66, 71, 86, 100, 114] is the presence of a single observer in the system that has access to all the uncoded measurements $y_{0:n}$. However, in sensor networks, each sensor acts as an observer. A time n, according to a given schedule, a particular sensor makes a measurement y_n, appropriately quantizes it to q_n and communicates it to the fusion center. Note that different sensors could use different measurement matrices. So, in general the measurement matrix H_n can vary with time. The fusion center uses the received quantized measurements $q_{0:n}$ to estimate the state x_n. Figure 2.1 outlines the overall filtering paradigm[1]. It is assumed that the sensors do not communicate between themselves. So, the quantized measurement q_n will be a function of the sensor's own analog measurement y_n and potential feedback from the fusion center. Unlike the classical case, there is no single entity in the network that has access to all the analog measurements $y_{0:n}$. Also, when a control input u_n is to be applied to the state-space process x_n, it is assumed that the fusion center determines u_n and applies the control input. So, we consider the setting where sensing takes place in a distributed manner but the controller is centralized.

In both cases above, the controller/fusion center needs to estimate the state using quantized measurements. Due to energy and bandwidth limitations, sensor networks provide a more compelling case for developing estimation algorithms using coarsely quantized measurements. Through most of the chapter, we focus only on estimation. Except in Section 2.6, where we study the separation between estimation and control, the control input u_n in (2.1) is assumed to be absent, i.e., $u_n = 0$.

[1]Here, we assume that the sensor communicates with the fusion center using a discrete rate-limited noiseless channel.

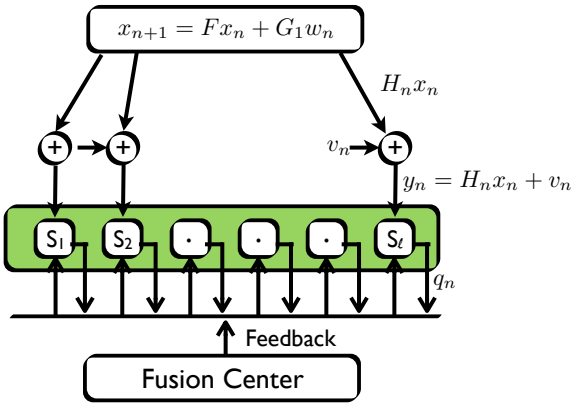


Figure 2.1: WSN with a fusion center: The sensors act as data gathering devices. S_i denotes the i^{th} sensor and in the above figure, S_ℓ is making the n^{th} measurement using the measurement matrix H_n.

2.2.2 Quantized Innovations and the Gaussian Assumption

A popular quantization scheme proposed for sensor networks is 'quantized innovations'. In this scheme, at each time n, the scheduled sensor makes the measurement y_n and also receives feedback from the fusion center in the form of a prediction $\hat{y}_{n|n-1} = \mathbb{E}y_n|q_{0:n-1}$. The sensor then quantizes its analog measurement y_n as $q_n = g(y_n - \hat{y}_{n|n-1})$ for some fixed finite quantizer $g(\cdot)$. Under the simplifying assumption that the prior $x_n|q_{0:n-1}$ is Gaussian, filtering equations of the following form have been obtained for $\hat{x}_{n|n} \triangleq \mathbb{E}x_n|q_{0:n-1}$ in [80, 113].

$$\hat{x}_{n|n} = \hat{x}_{n|n-1} + L\left(q_n\right) \frac{P_n H_n^T}{\left(H_n P_n H_n^T + R\right)^{1/2}}$$

$$\hat{x}_{n+1|n} = F_n \hat{x}_{n|n}$$

$$P_{n|n} = P_n - \lambda \frac{P_n H_n^T H_n P_n}{H_n P_n H_n^T + R} \tag{2.3a}$$

$$P_{n+1} \triangleq P_{n+1|n} = F_n P_{n|n} F_n^T + G_1 W G_1^T \tag{2.3b}$$

The value of λ and the mapping $L\left(q_n\right)$ depend on the quantization scheme used and are detailed in [113]. In particular, if $q_n = sign\left(y_n - \hat{y}_{n|n-1}\right)$, $\lambda = \frac{2}{\pi}$ and $L\left(q_n\right) =$

$\sqrt{\frac{2}{\pi}}q_n$. Eqs. (2.3a) and (2.3b) constitute the MLQ-Riccati with parameter λ. The above filter is optimal if the conditional distribution, $p\left(x_n|q_{0:n-1}\right)$, is Gaussian, which we will prove is generally a bad approximation. [98, 99] provide examples where the error performance of the filters in [80, 113] do not track the MLQ-Riccati that they were predicted to, i.e., Eq. (2.3). In order to understand the problem better, we take a closer look at the conditional law of $x_n|q_{0:n}$ in the following section. When $\{x_n\}$ and $\{y_n\}$ are jointly Gaussian, we will provide a novel stochastic characterization of x_n causally conditioned on the quantized measurement process $\{q_n\}$. This, in turn, allows us to identify the conditional density of $x_n|q_{0:n}$ to be, what we refer to as, a generalized closed skew normal distribution. We also use it to propose a novel filtering technique for the above problem which reduces to an elegant particle filter when $\{x_n\}$ and $\{y_n\}$ have linear state-space structure and outperforms the filters proposed in [80, 113], while providing much needed theoretical insight into the problem. Although the present work is motivated by sensor network applications, the results obtained are quite general as will become evident.

A note about the subscripts in F_n and H_n: In order to reduce notational clutter, in the rest of the chapter, we will drop the subscripts and just write F and H. In other words, we will present all results for the 'time invariant' case. The corresponding time varying versions can be obtained by simply replacing F (H) with F_n (H_n) wherever needed. The only exception to this rule is Corollary 2.4 which is applicable only to the time invariant case.

2.3 A Stochastic Characterization of the Conditional State Density

Suppose $\{x_n\}$ and $\{y_n\}$ are jointly Gaussian, then it is well known that the probability density of x_n conditioned on $y_{0:n}$ is a Gaussian with the following parameters

$$x_n|y_{0:n} \sim Z_n + R_{x_n y_{0:n}} R_{y_{0:n}}^{-1} y_{0:n} \quad \text{where} \tag{2.4}$$

$$Z_n \sim N_d(0, \underbrace{R_{x_n} - R_{x_n y_{0:n}} R_{y_{0:n}}^{-1} R_{y_{0:n} x_n}}_{\triangleq R^{\Delta}_{x_n, y_{0:n}}}) \tag{2.5}$$

When $\{x_n\}$ has an underlying state-space structure and $\{y_n\}$ is a linear measurement of $\{x_n\}$ corrupted by additive white Gaussian noise, as defined in Eq. (2.1), it is well known that the Riccati recursion in (2.2c) propagates the error covariance $P_n^{kf} \triangleq R^{\Delta}_{x_n, y_{0:n}} = \|x_n - \mathbb{E}x_n|y_{0:n}\|^2$. We would like to address the problem of optimal estimation using a quantized version of the observation process $\{y_n\}$. Let $\{q_n\}$ denote the quantized measurements obtained by causally quantizing $\{y_n\}$, i.e., q_n is a measurable function of $y_{0:n}$. We will show that the probability density of x_n conditioned on the quantized measurements $q_{0:n}$ admits a characterization very similar to Eq. (2.4). We state the result in the following Theorem.

Theorem 2.1. *The state x_n conditioned on the quantized measurements $q_{0:n}$ can be expressed as a sum of two independent random variables as follows*

$$x_n|q_{0:n} \sim Z_n + R_{x_n, y_{0:n}} R_{y_{0:n}}^{-1} \left[y_{0:n}|q_{0:n}\right], \quad \textit{where} \tag{2.6}$$

$$Z_n \sim N_d(0, R^{\Delta}_{x_n, y_{0:n}}) \tag{2.7}$$

Proof. See Appendix 2.9.1. □

Comparing Eqs. (2.4) and (2.6), the only difference is that the measurement vector $y_{0:n}$ has been replaced by the random variable $y_{0:n}|q_{0:n}$. It is easy to see that $y_{0:n}|q_{0:n}$ is a multivariate Gaussian random variable truncated to lie in the region defined by $q_{0:n}$. It is worth noting that the covariance of $x_n|q_{0:n}$, denoted by $\|x_n|q_{0:n}\|^2$, is given

by

$$\|x_n|q_{0:n}\|^2 = R^{\Delta}_{x_n,y_{0:n}} + R_{x_n,y_{0:n}} R^{-1}_{y_{0:n}} \|y_{0:n}|q_{0:n}\|^2 R^{-1}_{y_{0:n}} R_{y_{0:n},x_n} \tag{2.8}$$

Stating loosely, as the quantization scheme becomes finer, $y_{0:n}|q_{0:n}$ clearly converges to $y_{0:n}$ and $x_n|q_{0:n}$ approaches a Gaussian as is well known. Using Theorem 2.1, it is easy to see that $x_n|q_{0:n}$ is not Gaussian in general, contrary to the assumption made in [80,113]. Infact it belongs to a class of distributions, which we call the Generalized Closed Skew Normal Distributions (GCSN) (e.g., [35]), the details of which are given in the following section.

2.3.1 The Conditional State Distribution

Using Baye's rule, it is easy to see that

$$p(x_n|q_{0:n}) = p(x_n)\frac{p(q_{0:n}|x_n)}{p(q_{0:n})} = \phi_d(x_n; 0, R_{x_n})\frac{\Phi_n(\mathcal{S}_{q_{0:n}}; R_{y_{0:n},x_n}R^{-1}_{x_n}x_n, R^{\Delta}_{y_{0:n},x_n})}{\Phi_n(\mathcal{S}_{q_{0:n}}; 0, R_{y_{0:n}})} \tag{2.9}$$

$$R^{\Delta}_{y_{0:n},x_n} \triangleq R_{y_{0:n}} - R_{y_{0:n},x_n}R^{-1}_{x_n}R_{x_n,y_{0:n}}$$

where $\mathcal{S}_{q_{0:n}} \in \mathcal{B}(\mathbb{R}^n)$ is the region in which $y_{0:n}$ lies that is implied by a specific realization of the quantized measurements $q_{0:n}$. For example, consider the sign of innovation scheme, i.e., $q_n = sign(y_n - \hat{y}_{n|n-1})$. Then $q_n = 1$ implies that $y_n \in (\hat{y}_{n|n-1}, \infty)$, call this interval $\mathcal{S}_{n,q_{0:n}}$. Then, we can write $\mathcal{S}_{q_{0:n}}$ as $\mathcal{S}_{q_{0:n}} = \{y_{0:n} \in \mathbb{R}^{n+1} | y_i \in \mathcal{S}_{i,q_{0:i}},\ 0 \le i \le n\}$. The subscript $q_{0:n}$ is to emphasize that everything is conditioned on a fixed observation record, $q_{0:n}$.

The form of the distribution in (2.9) is very similar to what is studied in the statistics literature as the Closed Skew Normal distribution, which is defined as follows.

Definition 2.1 (Chapter 2, [35]). *Consider $d \ge 1$, $n \ge 1$, $\mu \in \mathbb{R}^d$, $\nu \in \mathbb{R}^n$, D an arbitrary $n \times d$ matrix, Σ and Δ positive definite matrices of dimensions $d \times d$ and $n \times n$ respectively. Then the probability density function of the closed skew normal*

distribution $CSN(\mu, \Sigma, D, \nu, \Delta)$ *is given by*

$$CSN(y; \mu, \Sigma, D, \nu, \Delta) = \phi_d(y; \mu, \Sigma) \frac{\Phi_n(-\infty, D(y-\mu); \nu, \Delta)}{\Phi_n(-\infty, 0; \nu, \Delta + D\Sigma D^T)} \tag{2.10}$$

Stochastically, $CSN(\mu, \Sigma, D, \nu, \Delta)$ is the probability density of X conditioned on the even $Z - D(X-\mu) < 0$, where $X \sim N_d(\mu, \Sigma)$ and $Z \sim N_n(\nu, \Delta)$ are independent and the inequality $Z - D(X-\mu) < 0$ is component-wise. One can arrive at this characterization by a simple application of Baye's rule. Skew normal distributions have generated a lot of interest ([4, 7, 35, 36, 60, 75]) because they provide a much needed tool to handle skewness in statistical modeling and have a good number of properties in common with the standard normal distribution, such as closure under marginalization and conditioning. In particular, such skew distributions arise via hidden truncation processes. In the context of estimation using quantized measurements, this truncation is the consequence of quantization, so such skew distributions naturally show up here. For example, consider the sign of innovation scheme given by $q_n = sign(y_n - \hat{y}_{n|n-1})$, where $\hat{y}_{n|n-1} = \mathbb{E}y_n|q_{0:n-1}$. In this setup, as will be shown below, the conditional law of $x_n|q_{0:n}$ is a closed skew normal distribution. Consider a fixed observation record $q_{0:n}$. Let $\xi_i = q_i y_i$ and $R_{\xi_{0:n}} = \text{diag}(q_{0:n}) R_{y_{0:n}} \text{diag}(q_{0:n})$. Then we have

$$\begin{aligned}
p(q_{0:n}) &= Pr\big(q_i(y_i - \hat{y}_{i|i-1}) \geq 0, \forall\ 0 \leq i \leq n\big) \\
&= \int\limits_{\substack{q_i y_i \geq q_i \hat{y}_{i|i-1} \\ 0 \leq i \leq n}} \cdots \int \phi_{n+1}\left(y_{0:n}; 0, R_{y_{0:n}}\right) dy_{0:n} \\
&= \int\limits_{\substack{\xi_i \geq q_i \hat{y}_{i|i-1} \\ 0 \leq i \leq n}} \cdots \int \phi_{n+1}\left(\xi_{0:n}; 0, R_{\xi_{0:n}}\right) d\xi_{0:n} \\
&= \Phi_{n+1}\left(-\infty, 0; \nu_n, R_{\xi_{0:n}}\right), \ \text{ where } \ \nu_n = [q_0 \hat{y}_{0|-1}, \ldots, q_n \hat{y}_{n|n-1}]^T
\end{aligned} \tag{2.11}$$

Similarly, one can show that

$$p(q_{0:n}|x_n) = \Phi_{n+1}\left(-\infty, R_{\xi_{0:n},x_n}R_{x_n}^{-1}x_n; \nu_n, R^{\Delta}_{\xi_{0:n},x_n}\right) \tag{2.12}$$

where $R_{\xi_{0:n},x_n} = \text{diag}(q_{0:n})R_{y_{0:n},x_n}$ and $R^{\Delta}_{\xi_{0:n},x_n} = R_{\xi_{0:n}} - R_{\xi_{0:n},x_n}R_{x_n}^{-1}R_{x_n,\xi_{0:n}}$. Using Eqs. (2.11) and (2.12), we get

$$\begin{aligned} p(x_n|q_{0:n}) &= p(x_n)\frac{p(q_{0:n}|x_n)}{p(q_{0:n})} \\ &= \phi_d\left(x_n; 0, R_{x_n}\right)\frac{\Phi_{n+1}\left(-\infty, R_{\xi_{0:n},x_n}R_{x_n}^{-1}x_n; \nu_n, R^{\Delta}_{\xi_{0:n},x_n}\right)}{\Phi_{n+1}\left(-\infty, 0; \nu_n, R_{\xi_{0:n}}\right)} \\ \implies p(x_n|q_{0:n}) &= CSN\left(x_n; 0, R_{x_n}, R_{\xi_{0:n},x_n}R_{x_n}^{-1}, \nu_n, R^{\Delta}_{\xi_{0:n},x_n}\right) \end{aligned} \tag{2.13}$$

In order to capture the effect of a general quantization scheme, one would need a straightforward generalization of the CSN distribution. It is obtained by considering the probability density of
$X|\left(Z - D(X-\mu) \in \mathcal{S}\right)$, where $\mathcal{S} \in \mathcal{B}(\mathbb{R}^n)$. This will result in probability density functions of the form (2.9). We will refer to such distributions as the generalized closed skew normal distributions (GCSN), which are formally defined as follows.

Definition 2.2. *For $x \in \mathbb{R}^n$ and $\mathcal{S} \in \mathcal{B}(\mathbb{R}^n)$, we define the generalized closed skew-normal distribution,*
$GCSN_{d,n}(x; \mu, \Sigma, D, \mathcal{S}, \Delta)$, as follows

$$\begin{aligned} GCSN_{d,n}(x; \mu, \Sigma, D, \mathcal{S}, \Delta) &\triangleq \phi_d\left(x; \mu, \Sigma\right) L_{d,n}(.) \\ L_{d,n}(.) &= \frac{\Phi_n\left(\mathcal{S}; D\left(x-\mu\right), \Delta\right)}{\Phi_n\left(\mathcal{S}; 0, \Delta + D\Sigma D^T\right)} \end{aligned} \tag{2.14}$$

Now, suppose $\{x_n\}$ and $\{y_n\}$ have the state-space structure of (2.1) and suppose W is positive definite for all $n \geq 0$. Then the evolution of the conditional state distribution with time is completely characterized by the following theorem.

Theorem 2.2 (Conditional State Distribution)**.** *The probability density function of $x_n|q_{0:n}$ is given by $GCSN_{d,n+1}\left(x_n; 0, R_{x_n}, R_{y_{0:n},x_n}R_{x_n}^{-1}, \mathcal{S}_{q_{0:n}}, R^{\Delta}_{y_{0:n},x_n}\right)$. The recursions*

relating the parameters of the distributions of $x_{n-1}|q_{0:n-1}$ *and* $x_n|q_{0:n}$ *are given by*

$$R_{x_n} = FR_{x_{n-1}}F^T + G_1WG_1^T,\ R_{y_{0:n},x_n} = \begin{bmatrix} R_{y_{0:n-1},x_n}F^T \\ H \end{bmatrix} \tag{2.15a}$$

$$R_{y_{0:n}} = \begin{bmatrix} R_{y_{0:n-1}} & R_{y_{0:n-1},x_n}A^TH^T \\ HFR_{x_n,y_{0:n-1}} & R + HR_{x_n}H^T \end{bmatrix},\ R_{y_0} = R + HR_{x_0}H^T \tag{2.15b}$$

$$\mathcal{S}_{q_{0:n}} = \mathcal{S}_{q_{0:n-1}} \cap \{y_n \in \mathcal{S}_{n,q_{0:n}}\} \tag{2.15c}$$

Proof. See Appendix 2.9.2. □

When the full measurements $y_{0:n}$ are available, the conditional state density is completely characterized by its mean and covariance which are propagated by the traditional Kalman filtering equations (Chapter 9, [50]). When only the quantized measurements are available, it is interesting to note that the conditional state distribution is completely characterized by a finite number of parameters which are propogated as given in Theorem 2.2. So, Eq. (2.15) constitutes the equivalent of the traditional Kalman filtering equations in the case when only the quantized measurements are available. In fact, one can write non-trivial formulae for the mean and covariance of a GCSN, but computing them will quickly become infeasible since the dimensions of some of the matrices involved in Eq. (2.15) grow with time. Except, $\mathcal{S}_{q_{0:n}}$, all other parameters are independent of the specific realization of the quantized measurements and hence, in principle, can be propagated offline. Theorem 2.1 can be used to translate any results on the properties of the closed skew normal distribution into additional insights on the current problem. Next we discuss a special case where we derive closed form Kalman-like recursions for the mmse estimate of the state and the corresponding estimation error.

2.3.2 A Comment on Quantizing the True Innovation

Suppose $\{x_n\}$ and $\{y_n\}$ have the linear state-space structure of (2.1) with $\{y_n\}$ being a scalar measurement process. The innovations process associated to $\{y_n\}$ is denoted

by $\{e_n\}$, i.e., $e_n = y_n - \mathbb{E}y_n|\mathbf{y}_{n-1}$ and $R_{e_n} \triangleq \|e_n\|^2$. The following notation shall be used in the rest of the chapter.

$$\hat{x}_{n|m} \triangleq Ex_n|q_{0:m},\ \hat{x}_n \triangleq \hat{x}_{n|n-1},\ \hat{x}^{kf}_{n|m} = Ex_n|y_{0:m},\ \hat{x}^{kf}_n \triangleq \hat{x}^{kf}_{n|n-1}$$
$$P_{n|m} \triangleq \|x_n - \hat{x}_{n|m}\|^2,\ P_n \triangleq P_{n|n-1}$$
$$P^{kf}_{n|m} \triangleq \|x_n - \hat{x}^{kf}_{n|m}\|^2,\ P^{kf}_n \triangleq P^{kf}_{n|n-1}$$

For ease of exposition, we assume a fixed quantizer $g(.)$ whose quantization intervals are given by $\{(z_0, z_1), (z_1, z_2), \ldots, (z_{\ell-1}, z_\ell), (z_\ell, z_{\ell+1})\}$, where $z_0 = -\infty$ and $z_{\ell+1} = \infty$. So, if $q_n = g\big(e_n/R^{1/2}_{e_n}\big)$, then a realization of $q_{0:n}$ would imply that $e_j/R^{1/2}_{e_j} \in (z_{l_j}, z_{l_j+1}),\ j \leq n$ for some $0 \leq l_j \leq \ell$. With this setup, we have the following result.

Theorem 2.3 (Optimal Estimation Using Quantized 'True' Innovations)**.** *The mmse estimate of x_n using $q_{0:n}$, denoted by $\hat{x}_{n|n}$, and the associated estimation error, denoted by $P_{n|n}$, are given recursively by the following equations*

$$\hat{x}_{n|n} = F\hat{x}_{n-1|n-1} + \frac{P^{kf}_n H^T}{\sqrt{HP^{kf}_n H^T + R}} \frac{\phi(z_{l_n}) - \phi(z_{l_n+1})}{\Phi(z_{l_n+1}) - \Phi(z_{l_n})} \tag{2.16a}$$
$$P_{n|n} = FP_{n-1|n-1}F^T - \alpha \frac{P^{kf}_n H^T H P^{kf}_n}{HP^{kf}_n H^T + R} + G_1 W G_1^T \tag{2.16b}$$
$$\alpha = \sum_{k=0}^{\ell} \frac{(\phi(z_k) - \phi(z_{k+1}))^2}{\Phi(z_{k+1}) - \Phi(z_k)},\ \ z_{\ell+1} \triangleq \infty,\ \ z_0 \triangleq -\infty \tag{2.16c}$$
$$P^{kf}_{n+1} = FP^{kf}_n F^T - \frac{FP^{kf}_n H^T H P^{kf}_n F^T}{HP^{kf}_n H^T + R} + G_1 W G_1^T \tag{2.16d}$$

Proof. See Appendix 2.9.3. □

Corollary 2.4 (Convergence of the Error Covariance)**.** *Suppose F is stable and Λ is the unique positive semidefinite solution to the discrete-time Lyapunov equation*

$$\Lambda = F\Lambda F^T + G_1 W G_1^T$$

and let P^{kf} be the unique positive semidefinite solution to the following discrete-time

algebraic Riccati equation (DARE)

$$Z = FZF^T - \frac{FZH^THZF^T}{HZH^T + R} + G_1WG_1^T$$

And let $P^f = P^{kf} - \dfrac{P^{kf}H^THP^{kf}}{HP^{kf}H^T + R}$. *Then the error covariance* $P_{n|n} \longrightarrow P$, *where* P *is given by*

$$P = \alpha P^f + (1 - \alpha)\Lambda \tag{2.17}$$

Further, if F *is unstable, then, irrespective of the quantization scheme used,* $P_{n|n} \longrightarrow \infty$.

Proof. See Appendix 2.9.4. □

For a fixed number of quantization levels, the value of α can be optimized by choosing $\{z_j\}_{j=1}^{\ell}$ appropriately. The above innovation coding scheme was introduced in [15] but closed form expressions for the optimal state estimate and the corresponding estimation error of the form stated above were not presented. The fact that $P_{n|n}$ diverges if F is unstable seems to be common knowledge (for eg, see [101]), the authors are not aware of a concrete proof before this work.

Note that the above scheme is not suited for distributed applications where no observer in the network has enough information to compute the innovations process. In general, the problem of optimal state estimation using quantized measurements does not admit an analytically tractable solution like the one above. This necessitates a numerical solution. But, using the insight of Theorem 2.1, we will show that $\hat{x}_{n|n}$ can be numerically approximated with a complexity that is, in most cases, comparable to the classical Kalman filter. In the following section, we outline the general particle filtering technique which will then be specialized to solve the problem of optimal state estimation using quantized measurements by exploiting Theorem 2.1.

2.4 The Kalman-Like Particle Filter

A promising approach to recursive estimation in nonlinear problems is particle filtering. For easy reference, a basic bootstrap filter for the case when $\{x_n\}$ and $\{y_n\}$ have state-space structure of (2.1) is outlined below. For example, if one uses the sign of innovation scheme, $q_n = sign(y_n - \hat{y}_{n|n-1})$, it is easy to see that the importance weights are given by $\omega_n^i = \Phi\left(q_n H(x_{n|n-1}^i - \hat{x}_{n|n-1}); 0, R\right)$. The particles in Alg 1 describe the conditional state density $p\left(x_n|q_{0:n}\right)$ and simulations suggest that one needs upwards of a thousand particles to get satisfactory error performance for most systems. In what follows, we use Theorem 2.1 to develop a novel particle filtering technique (KLPF) which converges to the optimal filter much faster than the generic filter outlined in Alg 1. The difference lies in using particles to describe a probability density with a much smaller covariance than the conditional state density. We begin by noting that

$$\mathbb{E}x_n|q_{0:n} = R_{x_n,y_{0:n}} R_{y_{0:n}}^{-1} \mathbb{E}y_{0:n}|q_{0:n} \tag{2.18}$$

So, it should suffice to propogate particles that are distributed as the random variable $\xi_n|q_{0:n}$ where

$$\xi_n = R_{x_n,y_{0:n}} R_{y_{0:n}}^{-1} y_{0:n} \tag{2.19}$$

The Kalman-like particle filter does exactly this, it propagates the conditional law $\xi_n|q_{0:n}$. Note that $\hat{x}_{n|n} = \mathbb{E}\xi_n|q_{0:n}$.

Recall that the quantizer output, q_n at time n, is obtained by quantizing a scalar-valued function of $y_n, q_{0:n-1}$. So, upon receiving q_n and using the previously received quantized values $q_{0:n-1}$, the fusion center infers that $y_n \in \mathcal{S}_{n,q_{0:n}}$ for some Borel measurable set $\mathcal{S}_{n,q_{0:n}}$.

In order to develop a particle filter to propogate $\xi_n|q_{0:n}$, one needs to understand the following evolution of the probability densities, $p(\xi_{n-1}|q_{0:n-1}) \rightarrow p(\xi_{n-1}|q_{0:n}) \rightarrow p(\xi_n|q_{0:n})$. We will begin by computing the likelihood ratio between $p(\xi_{n-1}|q_{0:n-1})$

Algorithm 1 Particle Filter

1. Set $n = 0$. Let $\{\alpha_M\}_{M\geq 1}$ be a nondecreasing sequence of positive integers[2]. For $i = 1, \cdots, M\alpha_M$, initialize the particles, $x^i_{0|-1} \sim p(x_0)$ and set $\hat{x}_{0|-1} = 0$

2. At time n, using measurement $q_n = g_n(y_{0:n})$, the importance weights are calculated as follows

$$\omega^i_n = p\left(q_n | x_n = x^i_{n|n-1}, q_{0:n-1}\right).$$

3. Measurement update is given by

$$\hat{x}^{pf,M}_{n|n} = \sum_{i=1}^{M\alpha_M} \overline{\omega}^i_n x^i_{n|n-1}$$

where $\overline{w}^i_n$ are the normalized weights, i.e.,

$$\overline{\omega}^j_n = \frac{\omega^j_n}{\sum_{i=1}^{M} \omega^i_n}$$

4. Resample M particles from the above $M\alpha_M$ particles with replacement as follows. Generate i.i.d random variables $\{J_\ell\}_{\ell=1}^{M}$, such that $P(J_\ell = i) = \overline{\omega}^i_n$. Then

$$x^\ell_{n|n} = x^{J_\ell}_n$$

5. For $i = 1, \cdots, M\alpha_M$, predict new particles according to,

$$\begin{aligned} &x^j_{n+1|n} \sim p\left(x_{n+1} | x_n = x^i_{n|n}, q_{0:n}\right), \text{ i.e.,} \\ &x^j_{n+1|n} = Fx^i_{n|n} + G_1 w^j_n, \ (i-1)\alpha_M + 1 \leq j \leq i\alpha_M \end{aligned}$$

where $\{w^j_n\}_{j=1}^{M\alpha_M}$ are sampled according to $p(w_n | x_n = x^i_n, q_{0:n})$. For the linear state-space model of (2.1), the process noise, w_n, is independent of the state, x_n, and the measurements, $q_{0:n}$. So, $\{w^j_n\}_{j=1}^{M\alpha_M}$ are just i.i.d $N_d(0, W)$.

6. Set $\hat{x}^{pf,M}_{n+1|n} = F\hat{x}^{pf,M}_{n|n}$. Also, set $n = n + 1$ and iterate from step 2.

and $p(\xi_{n-1}|q_{0:n})$.

Lemma 2.5 (Measurement Update)**.** *The likelihood ratio between the conditional laws of $\xi_{n-1}|q_{0:n}$ and $\xi_{n-1}|q_{0:n-1}$ is given by*

$$\frac{p(\xi_{n-1}|q_{0:n})}{p(\xi_{n-1}|q_{0:n-1})} \propto \Phi\left(\mathcal{S}_{n,q_{0:n}}; HF\xi_{n-1}, R_{e_n}\right) \tag{2.20}$$

Proof. See Appendix 2.9.5. □

So, if $\{\xi^i_{n-1|n-1}\}_i$ is a collection of particles distributed according to the law $p(\xi_{n-1}|q_{0:n-1})$. Then using Lemma 2.5, one can generate a new collection of particles $\{\xi^\ell_{n-1|n}\}_\ell$ that are distributed according to the law $p(\xi_{n-1}|q_{0:n})$ as follows. With each particle $\xi^i_{n-1|n-1}$, associate a weight $\omega^i = \Phi\left(\mathcal{S}_{n,q_{0:n}}; HF\xi^i_{n-1|n-1}, R_{e_n}\right)$. Generate i.i.d random variables $\{J_\ell\}_\ell$ such that $P(J_\ell = i) \propto \omega^i$ and set $\xi^\ell_{n-1|n} = \xi^{J_\ell}_{n-1|n-1}$. This is the standard resampling technique from steps (3) and (4) of Alg 1. Note that this amounts to a measurement update since we update the conditional law $p(\xi_{n-1}|q_{0:n-1})$ upon receiving the new measurement q_n.

Now consider the time update, i.e, going from $p(\xi_{n-1}|q_{0:n})$ to $p(\xi_n|q_{0:n})$. We will need the following result, the proof of which is simple.

Lemma 2.6. *The random variable $y_n|\xi_{n-1}, q_{0:n}$ is a truncated Gaussian and its probability density function is given by $\phi\left(\mathcal{S}_{n,q_{0:n}}; HF\xi_{n-1}, R_{e_n}\right)$.*

Observe that ξ_n is the mmse estimate of the state x_n given $y_{0:n}$. Since $\{x_n\}$ and $\{y_n\}$ have the state-space structure, it is well known that the Kalman filter propagates ξ_n recursively as follows

$$\xi_n = F\xi_{n-1} + K^f_n\big(y_n - HF\xi_{n-1}\big), \text{ where} \tag{2.21a}$$

$$K^f_n = \frac{P^{kf}_n H^T}{HP^{kf}_n H^T + R} \tag{2.21b}$$

Lemma 2.6 together with (2.21) completely describes the transition from $p(\xi_{n-1}|q_{0:n})$ to $p(\xi_n|q_{0:n})$. Taking cue from step 5) of Alg 1, suppose $\{\xi^\ell_{n-1|n}\}_\ell$ is a collection of particles distributed as $p(\xi_{n-1}|q_{0:n})$, then a new collection of particles $\{\xi^i_{n|n}\}_i$ that are

distributed as $p(\xi_n|q_{0:n})$ can be obtained as follows. For each $\xi^{\ell}_{n-1|n}$, generate $\{y^i_{n|n}\}$ for $(\ell-1)\alpha_M+1 \le i \le \ell\alpha_M$, i.i.d according to the law

$$p(y_n|\xi_{n-1}=\xi^{\ell}_{n-1|n}, q_{0:n}) = \phi\left(\mathcal{S}_{n,q_{0:n}}; HF\xi^{\ell}_{n-1|n}, R_{e_n}\right)$$

and set $\xi^i_{n|n} = F\xi^{\ell}_{n-1|n} + K^f_n\left(y^i_{n|n} - HF\xi^{\ell}_{n-1|n}\right)$.

Summarizing everything, we can desribe the Kalman-like particle filter as follows.

Algorithm 2 Kalman-Like Particle Filter (KLPF)

---

1. At $n=0$, generate $\{y^i_{0|0}\}_{i=1}^{M\alpha_M} \sim N(\mathcal{S}(q_0); 0, R_{y_0})$. Compute $\xi^i_{0|0} = K^f_0 y^i_{0|0}$

2. At time n, for each particle $\{\xi^i_{n-1|n-1}\}$, compute the weight as

$$\omega^i_n = \Phi\left(\mathcal{S}_{n,q_{0:n}}; HF\xi^i_{n-1|n-1}, R_{e_n}\right) \tag{2.22}$$

   Normalize the weights to get $\overline{\omega}^i_n = \frac{\omega^i_n}{\sum_{i=1}^{M\alpha_M}\omega^i_n}$

3. Resample M particles from the above $M\alpha_M$ particles with replacement as follows. Generate i.i.d random variables $\{J_\ell\}_{\ell=1}^{M}$, such that $P(J_\ell = i) = \overline{\omega}^i_n$. Then

$$\xi^{\ell}_{n-1|n} = \xi^{J_\ell}_{n-1|n-1}$$

4. Measurement update: Generate $y^i_{n|n}$ i.i.d from $\phi\left(\mathcal{S}_{n,q_{0:n}}; HF\xi^{\ell}_{n-1|n}, R_{e_n}\right)$, for $(\ell-1)\alpha_M+1 \le i \le \ell\alpha_M$ and obtain the new particles $\{\xi^i_{n|n}\}$ as follows

$$\xi^i_{n|n} = F\xi^{\ell}_{n-1|n} + K^f_n\left(y^i_{n|n} - HF\xi^{\ell}_{n-1|n}\right) \tag{2.23}$$

   The measurement updated estimate is given by

$$\hat{x}^{klpf,M}_{n|n} = \frac{1}{M\alpha_M}\sum_{i=1}^{M\alpha_M}\xi^i_{n|n} \tag{2.24}$$

5. Set $\hat{x}^{klpf,M}_{n+1|n} = F\hat{x}^{klpf,M}_{n|n}$. Also, set $n=n+1$ and iterate from step 2.

From (2.23) and (2.24), it is clear that the KLPF amounts to running $M\alpha_M$ Kalman filters in parallel that are driven by the measurements $\{y^i_{n|n}\}_{i=1}^{M\alpha_M}$ and taking

their average to get $\hat{x}_{n|n}^{klpf,M}$. This is why we refer to the filter as the Kalman-like particle filter.

2.4.1 KLPF Needs Fewer Particles

We will briefly argue why the KLPF needs fewer particles than the regular particle filter applied directly to the original problem. The particle filter outlined in Alg 1 propogates particles that at each time are distributed as $p(x_n|q_{0:n})$. The KLPF, on the other hand, propogates particles that are distributed as $p(\xi_n|q_{0:n})$. Recall from Theorem 2.1 that

$$x_n|q_{0:n} = Z_n + R_{x_n,y_{0:n}} R_{y_{0:n}}^{-1} [y_{0:n}|q_{0:n}] = Z_n + \xi_n|q_{0:n}$$

Further, since Z_n is independent of $\xi_n|q_{0:n}$, the covariance of $x_n|q_{0:n}$, $\|x_n|q_{0:n}\|^2$, is given by

$$\|x_n|q_{0:n}\|^2 = \|Z_n\|^2 + \|\xi_n|q_{0:n}\|^2 = P_{n|n}^{kf} + \|\xi_n|q_{0:n}\|^2$$

Hence, the covariance of $x_n|q_{0:n}$ is larger than that of $\xi_n|q_{0:n}$. In particular, as the number of quantization levels increases (appropriately), the covariance of $x_n|q_{0:n}$ converges to $P_{n|n}^{kf}$ while the covariance of $\xi_n|q_{0:n}$ converges to zero. As a result, with the same number of particles, the estimation error of the regular particle filter will be larger than that of the KLPF. Stated differently, for the same estimation performance, KLPF can do with far fewer particles. This can be substantiated mathematically by the following well known result in particle filtering literature.

Lemma 2.7 (Asymptotic Normality, e.g., Chapter 9 [21])**.** *Consider a scalar*[3] *linear state-space model* (2.1) *and let* $\hat{x}_{n|n} = \mathbb{E}x_n|q_{0:n}$. *Also suppose* $\lim_{M\to\infty} \alpha_M = +\infty$ *(e.g., choose* $\alpha_M = \log(M)$*), then asymptotically the normalized estimation error of Alg 1 and the KLPF converge, in distribution, to Gaussians whose variances are given*

[3]This result can be extended to vector-valued state-space models by applying the above lemma one component at a time.

as follows

$$\sqrt{M}\left(\hat{x}_{n|n}^{pf,M} - \hat{x}_{n|n}\right) \xrightarrow{\mathcal{D}} N\left(0, \|x_n - \hat{x}_{n|n}\|^2\right) \tag{2.25a}$$

$$\sqrt{M}\left(\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}\right) \xrightarrow{\mathcal{D}} N(0, \sigma_{n|n}^2), \ \textit{where} \tag{2.25b}$$

$$\sigma_{n|n}^2 \leq \|\xi_n - \hat{\xi}_{n|n}\|^2 = \|x_n - \hat{x}_{n|n}\|^2 - P_{n|n}^{kf} \tag{2.25c}$$

Simulations suggest that KLPF needs dramatically fewer particles as the quantization becomes finer. This will be demonstrated through examples in Section 2.7. Even for reasonably fine quantization, say 2 to 3 bits, $\|\xi_n - \hat{\xi}_{n|n}\|^2$ is much smaller than $\|x_n - \hat{x}_{n|n}\|^2$. In such examples, simulations suggest that the KLPF delivers close to optimal performance, i.e., $\left|\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}\right|$ is small with high probability, for $M \leq 100$.

Note that Lemma 2.7 does not provide any quantitative information about how many particles one would need to get a desired performance. In practice one would be interested in bounding $P\left(|\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}| > B\right)$ for finite M. All such results in the existing literature (e.g., [21, 22]) are available only for bounded functions of the state, i.e., for a bounded and appropriately well-behaved function $f(.)$, the behavior of $P\left(|f(\hat{x}_{n|n}^{klpf,M}) - f(\hat{x}_{n|n})| > B\right)$ is fairly well understood. Clearly functions of the form $f(x) = x$, which is what we are interested in, are not bounded and hence these results do not apply. In order for KLPF to be practically useful, one would need bounds on $\|x_n - \hat{x}_{n|n}\|^2$ and on $P\left(|\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}| > B\right)$ for finite M.

2.5 Consistency and Convergence of the KLPF

There is a vast body of literature on the convergence behavior of particle filters, [20–22, 104] being a representative sample. In this section, we will show that

$$\sqrt{M}\left(\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}\right) \quad \text{and} \quad \sqrt{M}\left(\hat{x}_{n|n}^{pf,M} - \hat{x}_{n|n}\right)$$

converge in distribution to zero mean Gaussian random variables. In particular, the former converges to a Gaussian random variable with a much smaller variance than the latter. For ease of exposition, we present all results for a scalar-valued state-space model, i.e., $x_n \in \mathbb{R}$. This can be extended to the vector case by treating x_n one component at a time and is straightforward. Most of the literature on the convergence of particle filters assumes the traditional measurement model, where the current measurement, conditioned on the current state, is independent of the past measurements. This is clearly not true for the quantization scheme we are considering. q_n is not undependent of $q_{0:n-1}$ conditioned on x_n. But the techniques themselves are quite general and can be easily extended to the more general measurement model at hand. Before presenting the convergence results on the particle filters proposed in the previous section, we need to introduce a couple of simple definitions. A sample of particles $\{z^i\}_{i=1}^M$ with associated weights $\{w^i\}_{i=1}^M$ is said to constitute a weighted sample $\{z^i, w^i\}_{i=1}^M$. For such a sample, consistency and asymptotic normality are defined as follows.

Definition 2.3 (Consistency)**.** *The weighted sample $\{(z^i, w^i)\}_{1\le i\le M}$ is said to be consistent for the probability measure ν and the set $\mathcal{C} \subseteq L^1(\mathbb{R}, \nu)$ if for any $f \in \mathcal{C}$,*

$$\sum_{i=1}^{M} \frac{w^i}{\sum_{j=1}^{M} w^j} f\left(z^i\right) \xrightarrow{P} \nu(f), \quad as \;\; M \to \infty$$

Definition 2.4 (Asymptotic Normality)**.** *Let $\mathbf{F}$ be a class of real-valued measurable functions on $\mathbb{R}$, let σ be a nonnegative function on $\mathbf{F}$, and let $\{\alpha_M\}$ be a nondecreasing real sequence diverging to infinity. We say that the weighted sample $\{(z^i, w^i)\}_{1\le i\le M}$ is asymptotically normal for $(\nu, \mathbf{F}, \sigma, \{\alpha_M\})$ if for any function $f \in \mathbf{F}$, it holds that $\nu(|f|) < \infty$, $\sigma^2(f) < \infty$ and*

$$\alpha_M \sum_{i=1}^{M} \frac{w^i}{\sum_{j=1}^{M} w^j} \left[f(z^i) - \nu(f)\right] \xrightarrow{\mathcal{D}} N\left(0, \sigma^2(f)\right), \quad as \quad M \longrightarrow \infty \tag{2.26}$$

In words, asymptotic normality implies that the estimation error is distributed as a zero-mean Gaussian with a fixed variance that is independent of the number

of samples, M, when M is large. Note that consistency follows from asymptotic normality.

We present the convergence results for the case $\alpha_M \to \infty$ since it allows a clean interpretation of the asymptotics. These can be extended to the more general case of $\alpha_M \to \alpha > 0$ at the expense of more involved notation without giving any additional insight into the problem. Also, if a measure ν admits a density p, we use ν and p interchangeably and the context would make it clear.

Theorem 2.8 (Weak convergence of Algorithm 1)**.** *The following holds true*

1. *If* $\{x^i_{0|-1}, 1\}_{i=1}^{M\alpha_M}$ *is consistent for* $\Big(p(x_0), L^1\big(\mathbb{R}, p(x_0)\big)\Big)$*, then for any* $n > 0$*,* $\{x^i_{n|n}\}_{i=1}^{M}$ *is consistent for* $\Big(p\left(x_n|\mathbf{q}_n\right), L^1\big(\mathbb{R}, p\left(x_n|\mathbf{q}_n\right)\big)\Big)$

2. *If in addition* $\{x^i(0|-1), 1\}_{i=1}^{M\alpha_M}$ *is asymptotically normal for* $\Big(p(x_0), L^2\big(\mathbb{R}, p(x_0)\big), Var_{p(x_0)}(.), \sqrt{M\alpha_M}\Big)$*, then for any* $n > 0$*,* $\{x^i_{n|n}\}_{i=1}^{M}$ *is asymptotically normal for* $\Big(p\left(x_n|\mathbf{q}_n\right), L^2\big(\mathbb{R}, p\left(x_n|\mathbf{q}_n\right)\big), Var_{p(x_n|\mathbf{q}_n)}(.), \sqrt{M}\Big)$*, in particular*

$$\sqrt{M}\left(\hat{x}^{pf,M}_{n|n} - \hat{x}_{n|n}\right) \xrightarrow{\mathcal{D}} N\left(0, \|x_n - \hat{x}_{n|n}\|^2\right) \tag{2.27}$$

In particular, whenever $\limsup_n \|x_n - \hat{x}_{n|n}\|^2 < \infty$, the above result implies that $\hat{x}^{pf,M}_{n|n} \to \hat{x}_{n|n}$ as $M \to \infty$.

Theorem 2.9 (Weak Convergence of Algorithm 2)**.** *The following holds true*

1. *If* $\{\xi^i_{0|0}, 1\}_{i=1}^{M\alpha_M}$ *is consistent for* $\Big(p(\xi_{0|0}), L^1\big(\mathbb{R}, p(\xi_{0|0})\big)\Big)$*, then for any* $n > 0$*,* $\{\xi^i_{n|n}\}_{i=1}^{M}$ *is consistent for* $\Big(p\left(\xi_n|\mathbf{q}_n\right), L^1\big(\mathbb{R}, p\left(\xi_n|\mathbf{q}_n\right)\big)\Big)$

2. *If in addition* $\{\xi^i_{0|0}, 1\}_{i=1}^{M\alpha_M}$ *is asymptotically normal for* $\Big(p(\xi_{0|0}), L^2\big(\mathbb{R}, p(\xi_{0|0})\big), Var_{p(\xi_{0|0})}(.), \sqrt{M\alpha_M}\Big)$*, then for any* $n > 0$*,* $\{\xi^i_{n|n}\}_{i=1}^{M\alpha_M}$ *is asymptotically normal for* $\Big(p\left(\xi_n|\mathbf{q}_n\right), L^2\big(\mathbb{R}, p\left(\xi_n|\mathbf{q}_n\right)\big), \sigma_{n|n}, \sqrt{M\alpha_M}\Big)$*, in partic-*

ular, for $f(x) = x$,

$$\sqrt{M}\left(\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}\right) \xrightarrow{\mathcal{D}} N(0, \sigma_{n|n}^2(f)), \text{ where} \tag{2.28}$$

$$\sigma_{n|n}^2(f) \leq \|\xi_n - \hat{\xi}_{n|n}\|^2 = R_{x_n,y_{0:n}} R_{y_{0:n}}^{-1} \|y_{0:n}|q_{0:n}\|^2 R_{y_{0:n}}^{-1} R_{y_{0:n},x_n} \tag{2.29}$$

Proofs for Theorem 2.8 and Theorem 2.9 follow from a straightforward extension of the results in Chapter 9 of [21]. Now, note that the asymptotic normality and consistency of $\{\xi_{0|0}^i\}$ and $\{x_{0|-1}^i\}$ follows from the fact that they are drawn i.i.d from $p(\xi_0|q_0)$ and $p(x_0)$, respectively. This observation, coupled with Theorem 2.8 and Theorem 2.9, proves the correctness of the brute force particlef filter and the KLPF. In addition to proving the correctness of the KLPF, Theorem 2.9 proves that the asymptotic variance of the estimates from Alg 2 is typically much smaller than that for Alg 1. The particles in the KLPF describe the random variable $R_{x_n,y_{0:n}} R_{y_{0:n}}^{-1} y_{0:n}|q_{0:n}$. Its variance decreases to zero as the number of quantization levels increases. On the other hand, the variance of $x_n|q_{0:n}$ cannot be smaller than $P_{n|n}^{kf}$. As a result KLPF needs dramatically fewer particles as the quantization becomes finer. This will be demonstrated through examples in Section 2.7. In practice, for most systems, $\|\xi_n - \hat{\xi}_{n|n}\|^2$ is much smaller than $\|x_n - \hat{x}_{n|n}\|^2$. In such examples, simulations suggest that the KLPF delivers close to optimal performance, i.e., $\left|\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}\right|$ is small with high probability, for $M \leq 100$. Though Theorems 2.8 and 2.9 prove the correctness and characterize the asymptotic behavior of the particle filters, there is more to be understood about the rates of convergence of the two algorithms. That is, in practice one would be interested in bounding $P\left(|\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}| > B\right)$ for finite M. All such results in the existing literature (e.g., [22]) are available only for bounded functions of the state. Clearly functions of the form $f(x) = x$, which is what we are interested in, are not bounded. Note that asymptotic normality only tells us that

$$P\left(\sqrt{M}|\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}| > B\right) \longrightarrow 2\Phi(B; 0, \sigma_{n|n}^2(f)), \text{ where } f(x) = x$$

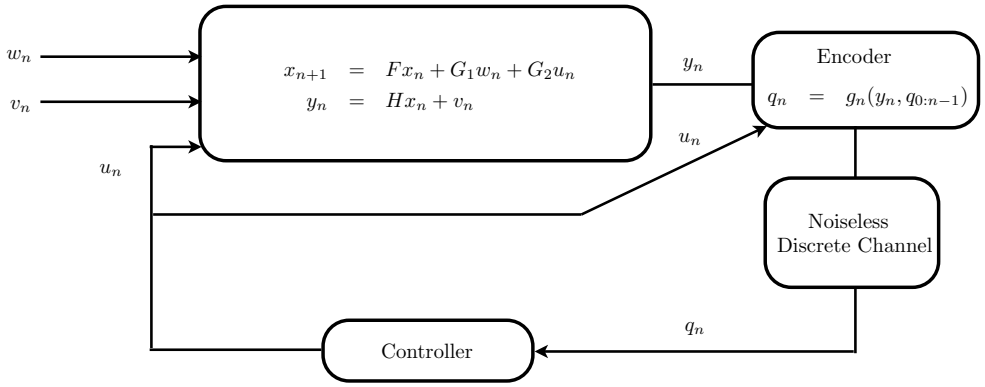


Figure 2.2: Measurement feedback control

In order to implement the KLPF in practice, one would need bounds on $\|x_n - \hat{x}_{n|n}\|^2$ and on
$P\left(|\hat{x}_{n|n}^{klpf,M} - \hat{x}_{n|n}| > B\right)$ for finite M.

2.6 The Separation Principle

Consider the closed-loop system outlined in Figure 2.2.

The traditional finite horizon linear quadratic Gaussian (LQG) problem [40] is one where the control input, u_n, is constrained to be a causal and linear function of the measurements $y_{0:n}$, i.e., $u_n = L_n(y_0, \ldots, y_n)$ for some linear function $L_n(.)$ (or $u_n = L_n(x_0, \ldots, x_n)$ in the full-information case) and the objective is to minimize a finite horizon quadratic cost function, which can be written as follows

$$\min_{\{L_n\}_{0\le n\le M}} \mathbb{E}_{\{x_0,\mathbf{w}_N,\mathbf{v}_N\}} J^c(N), \quad \text{where} \tag{2.30a}$$

$$J^c(N) = \sum_{n=0}^{N} \left[u_n^T M_u u_n + x_n^T M_x x_n\right] + x^T(N+1) M_o x(N+1) \tag{2.30b}$$

In the full-information case, it is well known that the optimum control action at time

n, u_n, depends only on the current state x_n and is given by (Chapter 9, [40])

$$u_n = -K_u x_n, \text{ where} \tag{2.31a}$$

$$K_u = (M_u + G_2^T M_o G_2)^{-1} G_2^T M_o A \tag{2.31b}$$

whereas in the case of measurement feedback, the optimal control is given by $u_n = -K_u \hat{x}^{kf}_{n|n}$, where $\hat{x}^{kf}_{n|n} = Ex_n|y_{0:n}$, which is linear in $y_{0:n}$ due to Gaussianity of the process and measurement noise[4]. Note that the control gain in the measurement feedback case is the same as in Eq. (2.31) and this is the well known separation principle (e.g., Chapter 9, [40]).

Consider the case when only the quantized measurements $\{q_n\}$ are available and the control action u_n is allowed to be a causal function (not necessarily linear) of the quantized measurements, i.e., $u_n = f_n(q_0, \ldots, q_n)$, where $f_n(.)$ is any function measurable w.r.t the sigma field generated by $q_{0:n}$. Consider the following control problem

$$\min_{\{f_n\}_{0 \le n \le M}} E_{\{x_0, \mathbf{w}_N, \mathbf{v}_N\}} J^c(N) \tag{2.32}$$

Note that the encoder/quantizer is fixed and the above minimization is over all possible control actions that are causal and measurable functions of the encoder outputs.

Theorem 2.10 (The Separation principle)**.** *The solution to (2.32) is given by the following certainty equivalent control law*

$$u_n = -K_u E x_n | q_{0:n} \tag{2.33}$$

where K_u, given by (2.31b), is the same control gain as in the full-information case.

Proof. The proof for this more general measurement model is a straightforward generalization of the proof presented in chapter 9, [40]. □

[4]In the absence of Gaussianity, $\hat{x}^{kf}_{n|n}$ would be the linear least-mean-squared estimate of x_n given $y_{0:n}$

Let $\hat{x}_{n|n} \triangleq Ex_n|q_{0:n}$ and $\tilde{x}_{n|n} \triangleq x_n - \hat{x}_{n|n}$. Then under the optimal control action, using the orthogonality of $\hat{x}_{n|n}$ and $\tilde{x}_{n|n}$, and simple algebra, EJ_n^c can be decomposed as follows

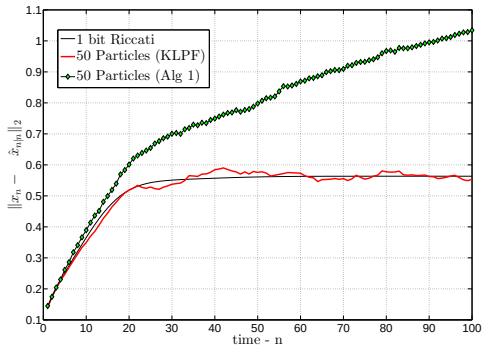
$$EJ_n^c = \underbrace{tr\left(M_o R_{x_{N+1}}\right) + \sum_{n=0}^{N} tr\left((K_u^T M_u K_u + M_x) R_{x_n}\right)}_{J_{LQ}} + \underbrace{E\tilde{x}_{N+1|N+1}^T M_o \tilde{x}_{N+1|N+1} + E\sum_{n=0}^{N} \tilde{x}_{n|n}^T M_x \tilde{x}_{n|n}}_{P_{e,N}^c} \tag{2.34}$$

J_{LQ} is the cost under full-state information and $P_{e,N}^c$ is the cost that depends on the estimation error covariance. So, the LQG problem of (2.32) reduces to minimizing $P_{e,N}^c$, completely decoupling estimation and control. Hence the problem of joint optimal estimation and control using quantized measurements reduces to one of finding the optimal causal encoding/quantization rule (see [16] for an interesting treatment of the optimal causal quantization problem). The separation result is not surprising and similar observations in the case of full-state information at the encoder were made in [101]. The separation principle equipped with the Kalman-like particle filter constitutes a computationally feasible framework to solve the optimal LQG problem using quantized measurements.

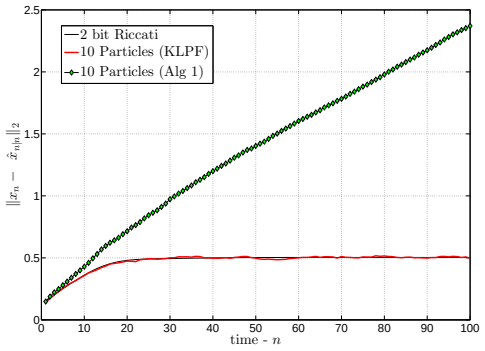
2.7 Simulations

The purpose of the following simulations is two fold, 1) to demonstrate that the KLPF needs far fewer particles than a naïve particle filter and 2) to demonstrate that the Gaussian assumption on the prior $p(x_n|q_{0:n-1})$, often used in the literature, can be quite inaccurate. Recall that in Alg 1, the particles describe the full probability density of the state conditioned on quantized measurements. While in the KLPF, part of the information about the conditional state density is captured neatly by the Kalman filter. So, the particles describe a truncated Gaussian which has a much smaller co-

variance than the conditional law of the state given the quantized observations. We give a few examples in this section to demonstrate the effectiveness of KLPF. We wrote the system matrices for all the examples in triangular form so that the eigen values can be easily read off from the diagonl entries.

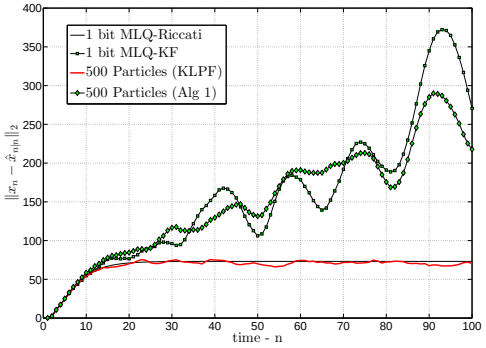


(a) 1 bit MLQ-KF, Alg 1 and KLPF

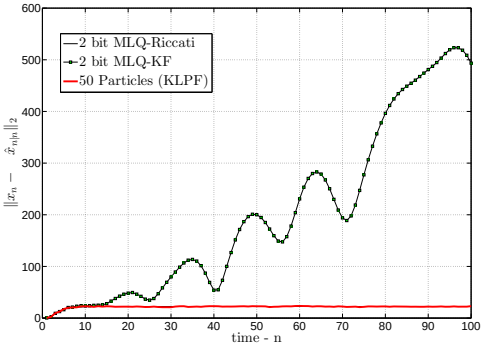


(b) 2 bit MLQ-KF, Alg 1 and KLPF

Figure 2.3: *Example 1: Both in (a) and (b), KLPF achieves good performance with remarkably few particles and hence has a complexity of the same order as that of a Kalman filter.*



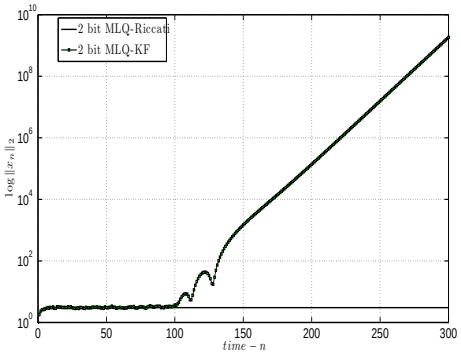
(a) 1 bit MLQ-KF, Alg 1 and KLPF



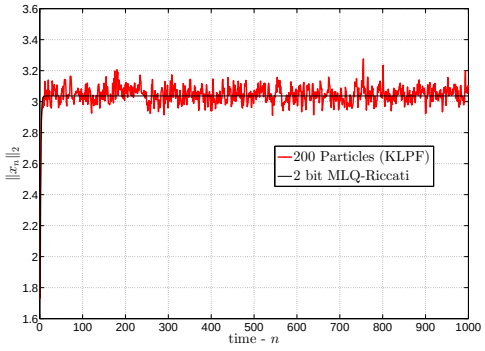
(b) 2 bit MLQ-KF, Alg 1 and KLPF

Figure 2.4: *Example 2: In (a), 1 bit MLQ-KF clearly diverges while KLPF converge to the optimal filter. From (b),2 bit MLQ-KF also diverges while KLPF performs well with just 50 particles. When using 2 bits, Alg 1 with 50 particles is orders of magnitude worse than KLPF and hence is not shown in the same plot*

In all the plots in this Section, '1-bit' stands for 'sign of innovation' and '2-bit' stands for a quantization rule with quantization intervals given by $(-\infty, -1.2437)$, $(-1.2437, -0.3823)$, $(-0.3823, 0.3823)$, $(0.3823, 1.2437)$ and $(1.2437, \infty)$. If the innovation falls in the interval $(-0.3823, 0.3823)$, no measurement update is done, so that



(a) MLQ-KF cannot control the system



(b) The KLPF can control the plant

Figure 2.5: Example 3: The plot for the KLPF has been shown over a longer time horizon of 1000 time instants to demonstrate convincingly that the KLPF can stabilize the unstable plant.

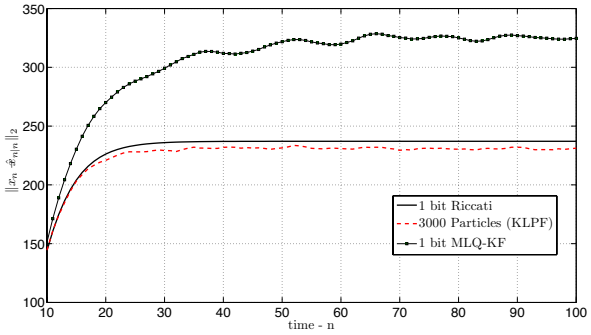


Figure 2.6: Example 4: Riccati is larger than the optimal error. This confirms that the optimal filter does not track the modified Riccati.

2 bits will suffice to represent the output of the above quantizer. The numbers in front of Alg 1 and KLPF denote the number of particles required to approximate the optimal filter closely. Clearly, KLPF requires far fewer particles than Alg 1. Also evident from Examples 1 and 2 is the fact that KLPF needs dramatically fewer particles as the quantization becomes finer.

Example 1: A simple tracking system can be characterized by the following parameters, $F = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $W = \begin{bmatrix} \frac{\tau^4}{4} & \frac{\tau^3}{2} \\ \frac{\tau^3}{2} & \tau^2 \end{bmatrix}$, $R = 0.81$ and $P_0 = 0.01\mathbb{I}_3$. Let the sampling period be $\tau = 0.1$. The plots are presented in Fig 2.3.

Example 2: Consider a linear time invariant system of the form (2.1) with the following parameters: $A = \begin{bmatrix} 0.95 & 1 & 0 \\ 0 & 0.9 & 10 \\ 0 & 0 & 0.95 \end{bmatrix}$, $h = \begin{bmatrix} 1 & 0 & 2 \end{bmatrix}$, $W = 2\mathcal{I}_3$, $R = 2.5$ and $P_0 =$

$0.01\mathbb{I}_3$, where $\mathbb{I}_m$ denotes an $m \times m$ identity matrix. Note that A is a stable matrix. As can be seen from Fig 2.4, 1 bit MLQ-KF and MLQ-KF diverge but KLPF delivers optimal performance with much fewer particles than Alg 1. With the addition of just 1 bit, the required number of particles drops from 500 to 50.

In Example 1, note that KLPF works with much fewer particles than in Example 2. One can attribute this to the much higher value of the optimal mean squared error in Example 2 than in Example 1, as can be seen from Figs 2.3 and 2.4.

Example 3 - Closing the loop: Here, we consider a system for which $x_{n+1} = Fx_n + w_n + u_n$ and $y_n = Hx_n + v_n$, where $F = \begin{bmatrix} 1.1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$, $w_n \sim N_3(0, \mathbb{I}_3)$ is the process noise, $v_n \sim N(0,1)$ is the observation noise and u_n is the control input. Also, consider the finite horizon quadratic cost function $\sum_{n=0}^{N} \|x_n\|^2$. Then the control policy that minimizes this cost is clearly $u_n = -F\hat{x}_{n|n}$. As seen from Fig 2.5, the 2 bit MLQ-KF fails to stabilize the system while KLPF stabilizes it with 100 particles.

Example 4: In [98], it was noted that the error performance of the optimal filter tracked the modified Riccati and it appeared that the modified Riccati is at least an upper bound on the error. This was investigated further with more examples and as seen in Figure 2.6, the optimal filter does not track the modified Riccati. This still leaves the possibility that the modified Riccati is an upper bound. Figure 2.6 corresponds to the system defined by $F = \begin{bmatrix} 0.95 & 1 & 1 & 0 & 0 \\ 0 & 0 & .9 & 7 & 1 \\ 0 & 0 & 0.6 & 2 & 0 \\ 0 & 0 & 0 & 0.7 & 0. \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 & 1 & 0 & 2 \end{bmatrix}$, $W = 2\mathcal{I}_5$, $R = 2.5$ and $P_0 = 0.01\mathbb{I}_5$

2.8 Summary

We propose a Kalman-like particle filter (KLPF) to optimally track and control a linear Gauss Markov process over a sensor network using quantized measurements. The technique is general and works for an arbitrary causal quantization scheme. In the examples studied, the KLPF required moderately small number of particles and therefore can obtain close to optimal performance with a computational complexity comparable to the conventional Kalman filter. We also showed that the classical separation principle between estimation and control holds. This allowed us to perform

optimal LQG control using quantized measurements.

An important open issue is to determine the number of particles necessary to closely approximate the optimal filter. In order to determine this, one needs upper bounds on the estimation error of the optimal filter and also understand the rate of convergence of particle filters. The error covariance matrix of the optimal filter seems to be upper bounded by the modified Riccati recursion introduced in [95]. Determining whether this is the case, and if so, why, remains an interesting open question. In particular, any meaningful upper bound on the estimation error of the optimal filter is necessary for practical applicability of the Kalman-like particle filter.

2.8.1 What If Communication Is Unreliable?

The focus of the present chapter has been on the optimal estimation of the state using quantized observations from a collection of sensors. The distributedness in this set up arises out of the absence of a single entity in the system which has access to all the *true* measurements. The primary focus is then on the quantization technique to be used at the sensors and the estimation technique to be used at the fusion center. In this setup, even though the communication between the sensors and the fusion center is rate limited, it is not noisy. But in practice, the data exchanged between the sensors and the fusion center is subject to errors. There could be many sources of such errors. It could be due to the communication channel itself, such as when the medium of communication is wireless which is true in most applications of practical interest. It could also be due to network congestion caused by competition for shared resources (e.g., an array of micro-actuators and sensors sharing a network). As a result, communication between different components of a networked control system (e.g., sensors and fusion center) can be fundamentally unreliable. Motivated by such a setup, we shift our focus to the problem of communicating data between various components while guaranteeing the *right kind of communication reliability*. It turns out that the notion of communication reliability prevalant in traditional communication systems is inadequate when the communication channel is present in

the feedback loop of a control system. In Chapter 3, we turn our attention to this interplay between the control and the communication problem.

2.9 Appendices

2.9.1 Proof of Theorem 2.1

The theorem will be proved by showing that the moment generating fuction of $x_n|q_{0:n}$ can be seen as the product of two moment generating functions corresponding to the two random variables in Eq. (2.6). Note that the moment generating function of a $d-$dim random variable X is given by $M_X(s) = \mathbb{E}e^{s^T X}$, $\forall\ s \in \mathbb{R}^d$.

$$p(x_n|q_{0:n}) = \int p(x_n, y_{0:n}|q_{0:n})dy_{0:n}$$

Noting that $p(x_n|y_{0:n}, q_{0:n}) = p(x_n|y_{0:n})$, we can write

$$\begin{aligned} Ee^{s^T x_n}|q_{0:n} &= \int e^{s^T x_n} p(x_n|y_{0:n})p(y_{0:n}|q_{0:n})dx_n dy_{0:n} \\ &\overset{(*)}{=} e^{\frac{1}{2}s^T R^{\Delta}_{x_n,y_{0:n}} s} \underbrace{\int e^{s^T R_{x_n,y_{0:n}} R^{-1}_{y_{0:n}} y_{0:n}} p(y_{0:n}|q_{0:n})dy_{0:n}}_{\triangleq \text{ mfg of } R_{x_n,y_{0:n}} R^{-1}_{y_{0:n}} y_{0:n}|q_{0:n}} \end{aligned}$$

$$\implies M_{x_n|q_{0:n}}(s) = M_{Z_n}(s) M_{y_{0:n}|q_{0:n}}(R^{-1}_{y_{0:n}} R_{y_{0:n},x_n} s) \tag{2.35}$$

where $Z_n \sim N_d(0, R^{\Delta}_{x_n,y_{0:n}})$. In getting $(*)$, we used the fact that

$$x_n|y_{0:n} \sim N_d(R_{x_n,y_{0:n}} R^{-1}_{y_{0:n}} y_{0:n}, R^{\Delta}_{x_n,y_{0:n}})$$

For any random variable Y, it is easy to see that $M_Y(A^T t) = M_{AY}(t)$. The result is now obvious from Eq. (2.35). Note that if $\{x_n\}$ and $\{y_n\}$ have the state-space structure, then $R^{\Delta}_{x_n,y_{0:n}} = P^{kf}_{n|n}$.

2.9.2 Proof of Theorem 2.2

$$\begin{aligned} p\left(x_n|q_{0:n}\right) &= p\left(x_n\right)\frac{p\left(q_{0:n}|x_n\right)}{p\left(q_{0:n}\right)} \\ &= p\left(x_n\right)\frac{p\left(y_{0:n}\in\mathcal{S}^{n+1}|x_n\right)}{p\left(q_{0:n}\right)} \\ &= \phi_d\left(x_n;0,R_{x_n}\right)\frac{\Phi_{n+1}\left(\mathcal{S}^{n+1};R_{y_{0:n},x_n}R_{x_n}^{-1}x_n,\Delta_n\right)}{\Phi_{n+1}\left(\mathcal{S}^{n+1};0,R_{y_{0:n}}\right)} \end{aligned}$$

Now $R_{y_{0:n},x_n} = \langle y_{0:n}, x_n\rangle = \left[\langle\mathcal{Y}_{n-1}, Ax_{n-1}+G_1w_n\rangle, \langle y_n, x_n\rangle\right]^T = \left[R_{x_n,y_{0:n-1}}, H^T\right]^T$. The recursion for $R_{y_{0:n}}$ follows similarly.

2.9.3 Proof of Theorem 2.3

Recall the definition of ξ_n from (2.19) and note that (2.21) propagates ξ_n. Recall that $\{e_n\}$ denotes the innovations process associated to the observation process $\{y_n\}$. So, $e_n = y_n - \mathbb{E}y_n|\mathbf{y}_{n-1} = y_n - HF\xi_{n-1}$. Now note that $\hat{x}_{n|n} \triangleq Ex_n|q_{0:n} = \mathbb{E}\xi_n|q_{0:n}$. So, from (2.21), we have

$$\hat{x}_{n|n} = F\mathbb{E}\xi_{n-1}|q_{0:n} + K_n^f\mathbb{E}e_n|q_{0:n}$$

Since q_i depends only on e_i that is independent of $e_j\ \forall\ i\neq j$, we have

$$\begin{aligned} \mathbb{E}\xi_{n-1}|q_{0:n} &= \mathbb{E}\xi_{n-1}|q_{0:n-1} = \hat{x}_{n-1|n-1} \quad \text{and} \\ \mathbb{E}e_n|q_{0:n} &= \mathbb{E}e_n|q_n = \mathbb{E}e_n|\left(e_n\in(z_{l_n}, z_{l_n+1})\right) \\ &= \|e_n\|_2\frac{\phi(z_{l_n})-\phi(z_{l_n+1})}{\Phi(z_{l_n+1})-\Phi(z_{l_n})} = \sqrt{HP_n^{kf}H^T+R}\frac{\phi(z_{l_n})-\phi(z_{l_n+1})}{\Phi(z_{l_n+1})-\Phi(z_{l_n})} \end{aligned}$$

So, we have

$$\hat{x}_{n|n} = F\hat{x}_{n-1|n-1} + \frac{P_n^{kf}H^T}{\sqrt{HP_n^{kf}H^T+R}}\frac{\phi(z_{l_n})-\phi(z_{l_n+1})}{\Phi(z_{l_n+1})-\Phi(z_{l_n})} \tag{2.36}$$

The corresponding error covariance is straightforward using orthogonality. One can rewrite (2.36) as

$$x_n - \hat{x}_{n|n} + \frac{P_n^{kf} H^T}{\sqrt{H P_n^{kf} H^T + R}} \frac{\phi(z_{l_n}) - \phi(z_{l_n+1})}{\Phi(z_{l_n+1}) - \Phi(z_{l_n})} = x_n - \hat{x}_{n-1|n-1}$$

Using orthogonality of $x_n - \hat{x}_{n|n}$ and $\frac{P_n^{kf} H^T}{\sqrt{H P_n^{kf} H^T + R}} \frac{\phi(z_{l_n}) - \phi(z_{l_n+1})}{\Phi(z_{l_n+1}) - \Phi(z_{l_n})}$, the result follows.

2.9.4 Proof of Corollary 2.4

Under the detectability and stabilizability assumptions, we know that $P_n^{kf} = \|x_n - Ex_n|\mathbf{y}_{n-1}\|^2$ converges to P^{kf}. Let P^f be the steady state value of $P_n^f \triangleq \|x_n - Ex_n|y_{0:n}\|^2$. Then

$$P^{kf} = F P^f F^T + G_1 W G_1^T$$
$$P^f = P^{kf} - \frac{P^{kf} H^T H P^{kf}}{H P^{kf} H^T + R}$$

Also, $\Lambda = F \Lambda F^T + G_1 W G_1^T$. Now let $B_n \triangleq P_{n|n} - \alpha P^f - (1-\alpha)\Lambda$ and $M_{f,n} \triangleq \frac{P_n^{kf} H^T H P_n^{kf}}{H P_n^{kf} H^T + R}$. Also let M_f denote the steady state value of $M_{f,n}$. Then from (2.16b), we have

$$\begin{aligned}
B_n &= F P_{n-1|n-1} F^T + G_1 W G_1^T - \alpha M_{f,n} - \alpha P^f - (1-\alpha)\Lambda \\
&= F P_{n-1|n-1} F^T + G_1 W G_1^T - \alpha M_{f,n} - \alpha \left(F P^f F^T + G_1 W G_1^T - M_f \right) \ldots \\
&\ldots - (1-\alpha) \left(F \Lambda F^T + G_1 W G_1^T \right) \\
&= F \left(P_{n-1|n-1} - \alpha P^f - (1-\alpha)\Lambda \right) F^T + \alpha \left(M_f - M_{f,n} \right) \\
&= F B_{n-1} F^T + \alpha \left(M_f - M_{f,n} \right)
\end{aligned}$$

Since $M_{f,n} \to M_f$, for each $\epsilon > 0$, there exists an M large enough such that $-\epsilon I \preceq M_f - M_{f,n} \preceq \epsilon I$ for all $n > N$. Then $B_n \to B$ and B satisfies (Lemma D.1.2 from [50])

$$-\epsilon \left(I + FF^T + F^2(F^T)^2 + \ldots\right) \preceq B \preceq \epsilon \left(I + FF^T + F^2(F^T)^2 + \ldots\right) \tag{2.37}$$

Since F is strictly stable and (2.37) is true for each $\epsilon > 0$, $B = \mathbf{0}$. If F is unstable, it is easy to see that $P_{n|n}$ diverges to infinity.

2.9.5 Proof of Lemma 2.5

An application of Baye's rule gives

$$\frac{p(\xi_{n-1}|q_{0:n})}{p(\xi_{n-1}|q_{0:n-1})} = \frac{P(q_n|q_{0:n-1}, \xi_{n-1})}{P(q_n|q_{0:n-1})} \propto P(q_n|q_{0:n-1}, \xi_{n-1})$$

Now, we have

$$\begin{aligned}
P(q_n|q_{0:n-1}, \xi_{n-1}) &= E\left[\left(\mathbb{I}_{y_n \in \mathcal{S}_{n,q_{0:n}}}\right)|q_{0:n-1}, \xi_{n-1}\right] \\
&= E\left[E\left(\mathbb{I}_{y_n \in \mathcal{S}_{n,q_{0:n}}}\right)|\mathbf{y}_{n-1}\right]|q_{0:n-1}, \xi_{n-1} \\
&= E\,\Phi\left(\mathcal{S}_{n,q_{0:n}}; HF\xi_{n-1}, R_{e_n}\right)|q_{0:n-1}, \xi_{n-1} \\
&= \Phi\left(\mathcal{S}_{n,q_{0:n}}; HF\xi_{n-1}, R_{e_n}\right)
\end{aligned}$$

Chapter 3

Sufficient Conditions for Closed-Loop Stability Over Noisy Channels

3.1 Introduction

At its simplest, control theory is concerned with regulating the behavior of dynamical systems using output feedback. A typical control system is comprised primarily of a dynamical system, an observer which measures the output of the dynamical system, a controller which uses the output to determine what feedback to apply and an actuator that applies the feedback determined by the controller. The controller needs to apply the control input in *real time*, and delay can result in loss of performance and/or instability. In most traditional control systems, the measurement and control subsystems are either colocated or hard-wired together and hence, there is no measurement loss. There are increasingly many applications on the horizon where we have systems that are remotely controlled over unreliable communication channels and networks. Broadly classified as cyber-physical systems, examples include the smart grid, distributed computation, intelligent highways, etc. (e.g., [70]). They are characterized by need to transmit measurement and control signals over noisy or bandwidth-limited channels. In such applications the conventional approach of using block coding to make the channels error-free is inappropriate as it introduces delay, which is anathema to the controller. On the other hand, purely control-theoretic

methods are also not appropriate because the measurements received by the controller are unreliable. Such a situation calls for a marriage of communication theory and control theory. The resulting area of control under communication constraints has received significant interest in recent years.

3.2 Background

Several aspects of the problem have been studied in the literature. In the context of rate-limited deterministic channels, significant progress has been made (e.g., [64, 66, 71]) in understanding the bandwidth requirements for stabilizing open-loop unstable systems. [62] considered feedback stabilization over stochastic communication channels where the stochasticity is modeled by a variable rate digital link and the encoder has causal knowledge of the number of bits transmitted error free. A result typical of this body of work can be described as follows. Consider the setup in Figure 3.1 where the noiseless digital channel allows up to R bits per time step of plant evolution on average. Then closed-loop stability is possible if and only if

$$R > \sum_{\lambda_i} \max\{0, \log|\lambda_i|\} \tag{3.1}$$

where $\{\lambda_i\}$ are the plant eigenvalues. The quantity on the right-hand side of (3.1) is often referred to as the intrinsic entropy rate of the plant.

One of the earliest papers to investigate the issues of communication constraints in control is [9] where the authors considered the problem of controlling plants over Gaussian channels with perfect feedback. Here perfect feedback implies that the channel encoder has causal knowledge of the channel outputs seen by the decoder. In such a setup, [9] showed that the encoder and the controller that minimize a quadratic cost are all linear and a more extensive treatment of this setup appears in [101]. In general, when the communication channel has continuous input and continuous output, has perfect feedback and imposes average power constraint, then it is possible to stabilize unstable systems over this channel using memoryless linear

encoders and controllers. Most channels in practice have discrete valued input and do not have perfect feedback, e.g., the internet that is best modeled as a packet erasure network.

Motivated by applications of networked control over packet erasure networks, [96] studied the problem of optimal LQG control of an open-loop unstable system when the measurements from the plant to the controller are subject to erasures and showed that closed-loop mean-squared stability is not possible if the probability of erasure exceeds a certain threshold. Similar results were obtained in [45, 67, 78, 90] in the case when the communication between the control unit and the actuation unit is prone to erasures. A result of this type is described as follows. Consider the case where the channel from the plant to the controller is error free but the control signals transmitted from the controller to the actuator are subject to Bernoulli erasures with erasure probability p. In such a setup, the state evolution is given by

$$x_{t+1} = Fx_t + Z_t u_t + w_t, \;\; t = 0, 1, \ldots$$

Here $\{Z_t\}$ is i.i.d Bernoulli$(1-p)$, i.e., $Z_t = 0$ with probability p and $Z_t = 1$ with probability $1-p$. Then $\mathbb{E}\|x_t\|^2$ grows unboundedly if and only if

$$\sqrt{p} > \frac{1}{\rho(F)}$$

where $\rho(F)$ is the spectral radius of F. So, if the erasure rate is high enough it is not possible to stabilize the system in closed-loop even with the optimal control law. This necessitates the need to encode the measurement and control signals to compensate for the channel errors. This is the purview of information and coding theory.

Shannon's single user information theory is concerned with reliable *one-way* communication of a message, that is available in its entirety, from a sender to a receiver over an unreliable channel. Reliability is achieved at the price of encoding-decoding delay. The focus is on communicating the message reliably and the associated delay is not of central concern. But in control systems, it is much more important to apply

an *approximate* feedback signal in *real-time* than to apply an *accurate* signal with a large *delay*. This is because feedback control systems are generally robust to such inaccuracies but are not as robust to delay (e.g., [56]). As a result, block encoding of the measurements is not applicable because the controller needs real time information about the system so that an appropriate control input can be applied. This is especially critical when the system being controlled is open-loop unstable. Any encoding-decoding delay translates into the system growing increasingly unstable. Consequently, a lot of literature is focused on stabilizing unstable systems since they accentuate the sensitivity of control systems to delay in the feedback loop.

It turns out that one needs the right trade-off between delay and accuracy in order to be able to stabilize unstable systems over noisy channels. Conventional notions of communication reliability such as block error probability are not compatible with this trade-off and consequently conventional error-correction techniques are inadequate. We will illustrate this trade-off through a simple example before continuing with the rest of the literature review followed by a chapter outline.

Anytime reliability through a toy example: Owing to the duality between estimation and control, the essential complexity of stabilizing an unstable process over a noisy communication channel can be captured by studying the open-loop estimation of the same process. We will motivate the kind of communication reliability needed for control through a simple example.

Example 3.1 (An Unstable Random Walk)**.** *Consider tracking the following random walk,*

$$x_{t+1} = \lambda x_t + w_t$$

where w_t takes values ± 1 with equal probability. Also $x_0 = 0$ and $|\lambda| > 1$. Suppose an observer observes x_t and communicates over a noisy communication channel to an estimator. Also assume that the estimator knows the system model and the initial state $x_0 = 0$. The objective then is for the estimator to track the state with an asymptotically bounded mean squared error.

The observer clearly needs to communicate whether w_t is 0 or 1. Note that the observer only has causal access to $\{w_i\}$, i.e., at any time t, the observer has access to $\{w_0, \ldots, w_{t-1}\}$. Let the encoding function of the observer at time t be $f_t : \mathbb{GF}_2^t \mapsto \mathcal{X}^n$, where $\mathcal{X}$ is the channel input alphabet and n is the number of channel uses available for each step of the system evolution. The encoding and decoding operations are depicted in Figure 3.3. Upon receiving the channel outputs until time t, the estimator generates estimates $\{\hat{w}_{0|t}, \hat{w}_{1|t}, \ldots, \hat{w}_{t-1|t}\}$ of the noise sequence $\{w_0, w_1, \ldots, w_{t-1}\}$. Then, the estimator's estimate of the state, $\hat{x}_{t+1|t}$, is given by

$$\hat{x}_{t+1|t} = \sum_{j=0}^{t} \lambda_{t-j} \hat{w}_{j|t} \tag{3.2}$$

Suppose $P^e_{d,t} = P\left(\text{argmin}_j(\hat{w}_{j|t} \neq \hat{w}_j) = t - d + 1\right)$, i.e., $P^e_{d,t}$ is the probability that the position of the earliest erroneous $\hat{w}_{j|t}$ is at time $j = t-d+1$. The probability here is over the randomness of the channel. From (3.2), we can bound $\mathbb{E}\left|x_{t+1} - \hat{x}_{t+1|t}\right|^2$ from above as

$$\begin{aligned}
&\sum_{w_{0:t}, \hat{w}_{0:t|t}} P\left(w_{0:t}, \hat{w}_{0:t|t}\right) \left| \sum_{j=1}^{n} \lambda^{t-j} (w_j - \hat{w}_{j|t}) \right|^2 \\
&\leq \sum_{d \leq t} P^e_{d,t} \left| \sum_{j=t-d+1}^{t} \lambda^{t-j} (w_j - \hat{w}_{j|t}) \right|^2 \\
&\leq \frac{4}{(|\lambda| - 1)^2} \sum_{d \leq t} P^e_{d,t} |\lambda|^{2d}
\end{aligned}$$

Clearly, a sufficient condition for $\limsup_t \mathbb{E}\left|x_{t+1} - \hat{x}_{t+1|t}\right|^2$ to be finite is as follows

$$P^e_{d,t} \leq |\lambda|^{-(2+\delta)d} \ \forall \ d \geq d_o, \ \ t > t_o \ \text{ and } \delta > 0 \tag{3.3}$$

where d_o and t_o are constants that do no depend on t, d. Any encoder-decoder pair that guarantees a reliability of the type (3.3) is said to be *anytime reliable*. We will define it more precisely in Section 3.4. In the example above, we need to communicate one information bit for each step of the plant evolution and this does not depend on

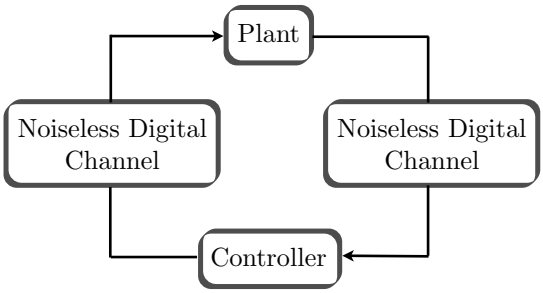
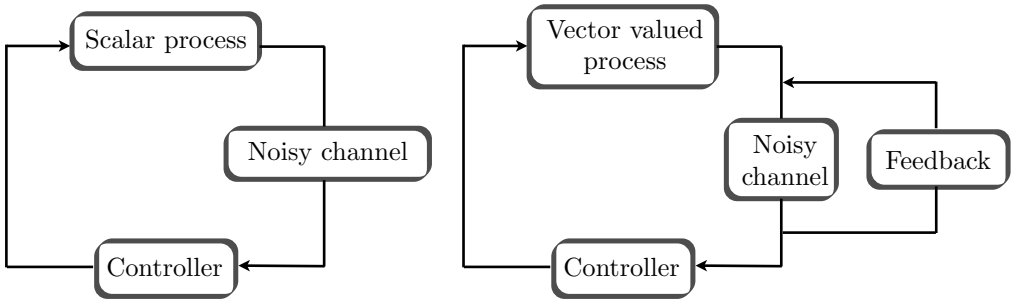


Figure 3.1: Stabilizing systems over noiseless digital channels with a data rate limit



(a) Stabilizing scalar processes without channel feedback

(b) Stabilizing vector-valued processes with channel feedback

Figure 3.2: Anytime capacity is the right notion for stabilizing systems over noisy channels

the system eigen value λ. This is an artifact of the discrete noise model in which noise takes only two possible values. For more common noise models, the number of information bits that need to be communicated in each time step will depend on λ.

In the context of control, it was first observed in [86] that exponential reliability of the form (3.3) is required to stabilize unstable plants over noisy communication channels and the notion of anytime reliability was introduced as the appropriate measure of communication reliability for channels that are in the feedback loop of control systems. Furthermore, [86] and [87] presented sufficient conditions on the rate of communication required and the size of the exponent in the exponential decay of $P_{t,d}^{e}$ for closed-loop stability for the scenarios depicted in Figures 3.2(a) and 3.2(b), respectively.

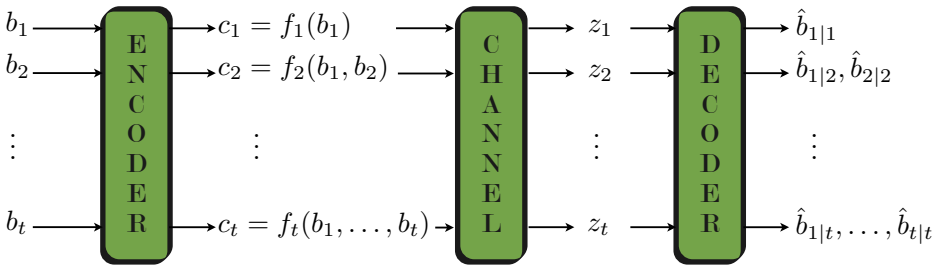


Figure 3.3: Causal encoding and decoding

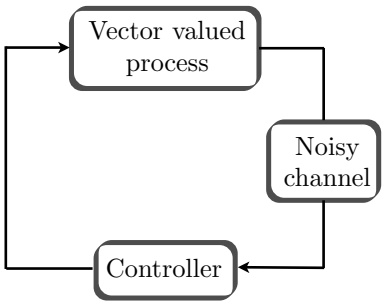


Figure 3.4: Stabilizing systems over noisy channels without channel feedback

3.3 Outline

In this chapter, we present novel nonasymptotic sufficient conditions for stabilizing vector-valued state-space processes over noisy channels for the setup shown in Figure 3.4. In Section 3.5, we present sufficient conditions for the case of scalar-valued measurements and in Section 3.6 we treat the vector case. We discuss the results and compare them with those in the literature in Section 3.7.

3.4 Problem Setup

The notation to be used in the rest of the Chapter is summarized in Table 3.1. Consider the following m_x-dimensional unstable linear system with m_y-dimensional measurements. Assume that (F, H) is observable and (F, G) is controllable.

$$x_{t+1} = Fx_t + Gu_t + w_t, \quad y_t = Hx_t + v_t \tag{3.4}$$

Table 3.1: Notation for Chapter 3

$H(.)$	The binary entropy function
$H^{-1}(y)$	The smaller root of the equation $H(x) = y$
For a matrix F, $\overline{F}$	abs(F), i.e., $\overline{F}_{i,j} = \lvert F_{i,j}\rvert . \forall\ i, j$
$\rho(F)$	Spectral radius of F
For a vector x, $x^{(i)}$	The i^{th} component of x
$\mathbf{1}_m$	$[1, \ldots, 1]^T$, i.e., a column with m 1's
For $w, v \in \mathbb{R}^m$, $w \gneqq v$	Component-wise inequality
$\log(.)$	Logarithm in base 2
For $0 \leq x, y \leq 1$, $KL\,(x\|y)$	$x \log \dfrac{x}{y} + (1-x) \log \dfrac{1-x}{1-y}$, i.e., Kullbeck-Leibler divergence between Bernoulli(x) and Bernoulli(y)

where

$$F = \begin{bmatrix} -a_1 & 1 & 0 & \ldots & \\ -a_2 & 0 & 1 & 0 & \\ \vdots & \vdots & & \ddots & \\ -a_{m-1} & \ldots & \ldots & 0 & 1 \\ -a_m & 0 & \ldots & \ldots & 0 \end{bmatrix}, \quad H = [1, 0, \ldots, 0]$$

where $\rho(F) > 1$, u_t is the m_u-dimensional control input and, w_t and v_t are bounded process and measurement noise variables, i.e., $\|w_t\|_\infty < \frac{W}{2}$ and $\|v_t\|_\infty < \frac{V}{2}$ for all t. Note that we do not make any distributional assumptions on the noise. The measurements $\{y_t\}$ are made by an observer while the control inputs $\{u_t\}$ are applied by a remote controller that is connected to the observer by a noisy communication channel. We assume that the control input is available to the plant losslessly. We do not assume that the observer has access to either the channel outputs or the control inputs. As is shown to be possible, e.g., in [64, 86], we do not use the control actions to communicate the channel outputs back to the observer through the plant because this could have a detrimental effect on the performance of the controller.

Before proceeding further, a word is in order about the boundedness assumption

on the noise. If the process and/or measurement noise have unbounded support, it is not clear how one can stabilize the system without additional assumptions on the channel. For example, [115] assumes feedback of channel outputs to the observer in order to stabilize an unstable process perturbed by Gaussian noise over an erasure channel while [118] proposes a forward side channel between the observer and the controller that has a positive zero error capacity. We avoid this difficulty by assuming that the noise has bounded support which may be a reasonable assumption to make in practice.

The measurements $y_{0:t-1}$ will need to be quantized and encoded by the observer to provide protection from the noisy channel while the controller will need to decode the channel outputs to estimate the state x_t and apply a suitable control input u_t. This can be accomplished by employing a channel encoder at the observer and a decoder at the controller. For simplicity, we will assume that the channel input alphabet is binary. Suppose one time step of system evolution in (3.4) corresponds to n channel uses[1], i.e., n bits can be transmitted for each measurement of the system. Then, at each instant of time t, the operations performed by the observer, the channel encoder, the channel decoder and the controller can be described as follows. The observer generates a $k-$bit message, $b_t \in \mathbb{GF}^k$, that is a causal function of the measurements, i.e., it depends only on $y_{0:t}$. Then the channel encoder causally encodes $b_{0:t} \in \mathbb{GF}^{kt}$ to generate the n channel inputs $c_t \in \mathbb{GF}^n$. Note that the rate of the channel encoder is $R = k/n$. Denote the n channel outputs corresponding to c_t by $z_t \in \mathcal{Z}^n$, where $\mathcal{Z}$ denotes the channel output alphabet. Using the channel outputs received so far, i.e., $z_{0:t} \in \mathcal{Z}^{nt}$, the channel decoder generates estimates $\{\hat{b}_{\tau|t}\}_{\tau \le t}$ of $\{b_\tau\}_{\tau \le t}$, which, in turn, the controller uses to generate the control input u_{t+1}. This is illustrated in Fig. 3.3.

With this setup, we can define the notion of anytime reliability as follows

Definition 3.1 (Anytime reliability)**.** *Given a channel that can carry n bits of*

[1]In practice, the system evolution in (3.4) is obtained by discretizing a continuous time differential equation. So, the interval of discretization could be adjusted to correspond to an integer number of channel uses, provided the channel use instances are close enough.

data for each time step of plant evolution, we say that an encoder-decoder pair is $(R, \beta, d_o)-$*anytime reliable over this channel if*

$$P^e_{t,d} \leq 2^{-n\beta d}, \ \ \forall \ t, d \geq d_o \tag{3.5}$$

In some cases, we write that a code is $(R, \beta)-$*anytime reliable. This means that there exists a fixed* $d_o > 0$ *such that the code is* $(R, \beta, d_o)-$*anytime reliable.*

Note that the exponent β is normalized with respect to the number of data bits n that the channel can carry in each time step. For example, if the channel carries one symbol per time step and the channel input alphabet has cardinality, say m, then we set $n = \log m$. We adopt this convention because we do not want the bounds on the rate and exponent that we will compute in Chapter 4 to depend on n.

We will show in Sections 3.5 and 3.6 that $(R, \beta)-$anytime reliability with an appropriately large rate, R, and exponent, β, is a sufficient condition to stabilize (3.4) in the mean-squared sense[2].

3.5 Sufficient Conditions for Stabilization — Scalar Measurements

Recall that we do not assume any feedback about the channel outputs or the control inputs at the observer/encoder. This is the setup we imply whenever we say that no feedback is assumed. In this context [86] derives a sufficient condition for stabilizing scalar linear systems over noisy channels without feedback while [87] considers stabilizing vector-valued processes in the presence of feedback. So, to the best of our knowledge, there are no results on stabilizing unstable vector-valued processes over a noisy channel when the observer does not have access to either the control inputs or the channel outputs.

We will develop two sufficient conditions for stabilizing vector-valued processes over noisy channels without feedback. The two sufficient conditions are based on two

[2]can be easily extended to any other norm

different estimation algorithms employed by the controller and neither is stronger than the other. We will then show in Section 3.7.1 that both sufficient conditions are asymptotically tight. For ease of presentation, we will treat the case of scalar and vector measurements separately. We will present the sufficient conditions for the case of scalar measurements here while vector measurements will be treated in Section 3.6

Consider the unstable m_x-dimensional linear state-space model in (3.4) with scalar measurements, i.e., $\rho(F) > 1$, and $m_y = 1$. Suppose that the characteristic polynomial of F is given by

$$f(z) \triangleq z^{m_x} + a_1 z^{m_x-1} + \ldots + a_{m_x}$$

Without loss of generality we assume that (F, H) are in the following canonical form.

$$F = \begin{bmatrix} -a_1 & 1 & 0 & \ldots & \\ -a_2 & 0 & 1 & 0 & \\ \vdots & \vdots & & \ddots & \\ -a_{m-1} & \ldots & \ldots & 0 & 1 \\ -a_m & 0 & \ldots & \ldots & 0 \end{bmatrix}, \quad H = [1, 0, \ldots, 0]$$

Owing to the duality between estimation and control, we can focus on the problem of tracking (3.4) over a noisy communication channel. For, if (3.4) can be tracked with an asymptotically finite mean-squared error and if (F, G) is stabilizable, then it is a simple exercise to see that there exists a control law $\{u_t\}$ that will stabilize the plant in the mean-squared sense, i.e., $\limsup_t \mathbb{E}\|x_t\|^2 < \infty$. In particular, if the control gain K is chosen such that $F + GK$ is stable, then $u_t = K\hat{x}_{t|t}$ will stabilize the plant, where $\hat{x}_{t|t}$ is the estimate of x_t using channel outputs up to time t. In control parlance, this amounts to verifying that the control input does not have a *dual effect* [10]. Hence, in the rest of the analysis, we will focus on tracking (3.4). The control input u_t therefore is assumed to be absent, i.e., $u_t = 0$.

3.5.1 Hypercuboidal Filter

We bound the set of all possible states that are consistent with the estimates of the quantized measurements using a hypercuboid, i.e., a region of the form

$$\{x \in \mathbb{R}^{m_x} | \mathbf{a} \leq x \leq \mathbf{b}\}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{m_x}$ and the inequalities are component-wise.

Since we assume that the initial state x_0 has bounded support, we can write $x_{min,0|-1} \leq x_0 \leq x_{max,0|-1}$ and suppose using the channel outputs received till time $t-1$, we have $x_{min,t|t-1} \leq x_t \leq x_{max,t|t-1}$. Since $H = [1, 0, \ldots, 0]$, the measurement update provides information of the form $x^{(1)}_{min,t|t} \leq x^{(1)}_t \leq x^{(1)}_{max,t|t}$ while there will be no additional information on other components of x_t. Note that an estimate of the state is given by the midpoint of this region, i.e., $\hat{x}_{t|t} = 0.5(x_{min,t|t} + x_{max,t|t})$. If we define $\Delta_{t|t} = x_{max,t|t} - x_{min,t|t}$, then the estimation error is asymptotically bounded if every component of $\Delta_{t|t}$ is asymptotically bounded. Using such a filter, we can stabilize the system in the mean-squared sense over a noisy channel provided that the rate R and exponent β of the $(R, \beta)-$anytime reliable code used to encode the measurements satisfy the following sufficient condition

Theorem 3.1. *It is possible to stabilize* (3.4) *in the mean-squared sense with an* $(R, \beta)-$*anytime code provided*

$$R > R_n = \frac{1}{n} \log_2 \sum_{i=1}^{m_x} |a_i|, \quad \beta > \beta_n = \frac{2}{n} \log_2 \rho(\overline{F}) \tag{3.6}$$

Proof. See Appendix 3.9.1 □

Before proceeding further, we will provide a brief sketch of the proof. Note that $\Delta_{t|t} = x_{max,t|t} - x_{min,t|t}$ is a measure of the uncertainty in the state estimate. From Lemma 3.7, $\Delta_{t+1|t} = \overline{F}\Delta_{t|t} + W\mathbf{1}_{m_x}$. The anytime exponent is determined by the growth of Δ_t in the absence of measurements, hence the bound $\beta_n = 2\log_2 \rho(\overline{F})$. The bound on the rate is determined by how fine the quantization needs to be for Δ_t to be

bounded asymptotically. It will be shown in Section 3.9.5 that $\rho\left(\overline{F}\right)$ is always larger than $\rho\left(F\right)$. By using an alternate filtering algorithm, which we call the Ellipsoidal filter, one can improve this requirement on the exponent from $\beta_n > 2\log_2 \rho(\overline{F})$ to $\beta_n > 2\log_2 \rho(F)$. But this will come at the price of a larger rate.

3.5.2 Ellipsoidal Filter

One can alternately bound the set of all possible states that are consistent with the estimates of the quantized measurements using an ellipsoid

$$\mathcal{E}(P,c) \triangleq \left\{x \in \mathbb{R}^{m_x} | \langle x-c, P^{-1}(x-c)\rangle \leq 1\right\}$$

This can be seen as an extension of the technique proposed in [93] to filtering using quantized measurements. If $m_x = 1$, $\rho(\overline{F}) = \rho(F)$. So, let $m_x \geq 2$.

Let $x_0 \in \mathcal{E}(P_0, 0)$ and suppose using the channel outputs received till time $t-1$, we have $x_t \in \mathcal{E}(P_{t|t-1}, \hat{x}_{t|t-1})$. Since $H = [1, 0, \ldots, 0]$, the measurement update provides information of the form $x^{(1)}_{min,t|t} \leq x^{(1)}_t \leq x^{(1)}_{max,t|t}$, which one may call a slab. $\mathcal{E}(P_{t|t}, \hat{x}_{t|t})$ would then be an ellipsoid that contains the intersection of the above slab with $\mathcal{E}(P_{t|t-1}, \hat{x}_{t|t-1})$, in particular one can set it to be the minimum-volume ellipsoid covering this intersection. Lemma 3.9 gives a formula for the minimum-volume ellipsoid covering the intersection of an ellipsoid and a slab. For the time update, it is easy to see that for any $\epsilon' > 0$ and $P_{t+1} = (1+\epsilon')FP_{t|t}F^T + \frac{W^2}{4\epsilon'}\mathbf{1}_{m_x}\mathbf{1}^T_{m_x}$, $\mathcal{E}(P_{t+1}, F\hat{x}_{t|t})$ contains the state x_{t+1} whenever $\mathcal{E}(P_{t|t}, \hat{x}_{t|t})$ contains x_t. This leads to the following lemma, the proof of which is contained in the discussion above. For convenience, we write P_t for $P_{t|t-1}$.

Lemma 3.2 (The Ellipsoidal Filter)**.** *Whenever $\mathcal{E}(P_0, 0)$ contains x_0, for each $\epsilon' > 0$, the following filtering equations give a sequence of ellipsoids $\left\{\mathcal{E}(P_{t|t}, \hat{x}_{t|t})\right\}$ that, at each*

time t, contain x_t.

$$P_{t+1} = (1+\epsilon')FP_{t|t}F^T + \frac{W^2}{4\epsilon'}\mathbf{1}_{m_x},\ \hat{x}_{t+1} = F\hat{x}_{t|t} \tag{3.7a}$$

$$P_{t|t} = b_tP_t - (b_t - a_t)\frac{P_te_1e_1^TP_t}{e_1^TP_te_1},\ \hat{x}_{t|t} = \xi_t\frac{P_te_1}{\sqrt{e_1^TP_te_1}} \tag{3.7b}$$

where a_t, b_t and ξ_t can be calculated in closed form using Lemma 3.9, and e_1 is the m_x-dimensional unit vector $e_1 = [1, 0, \ldots, 0]^T$.

Using this approach, we get the following sufficient condition.

Theorem 3.3. *It is possible to stabilize* (3.4) *for $m_x \geq 2$ in the mean-squared sense with an $(R, \beta)-$anytime code provided*

$$R > R_{e,n} = \frac{1}{n}\log_2\left[\sqrt{m_x}\sum_{i=1}^{m_x}|a_i|\theta^{i-1}\right] \tag{3.8a}$$

$$\beta > \beta_{e,n} = \frac{2}{n}\log_2\rho(F) \tag{3.8b}$$

where $\theta = \sqrt{\frac{m_x}{m_x-1}}$

Proof. See Appendix 3.9.4 □

3.6 Sufficient Conditions for Stabilization — Vector Measurements

As in the scalar case, we will assume without loss of generality that (F, H) are in a canonical form (is obtained from a simple transformation of *Scheme I* in Section 6.4.6 of [49]) with the following structure. F is a $q \times q$ block lower triangular matrix with $F^{i,j}$ denoting the $(i,j)^{th}$ block. So, $F^{i,j} = 0$ if $j > i$. $F^{i,j}$ is an $\ell_i \times \ell_j$ matrix

and $\sum_{i=1}^{q} \ell_i = m_x$. The diagonal blocks $F^{i,i}$ have the following structure.

$$F^{i,i} = \begin{bmatrix} -a_{i,1} & 1 & 0 & \dots & \\ -a_{i,2} & 0 & 1 & 0 & \\ \vdots & \vdots & & \ddots & \\ -a_{i,\ell_i-1} & \dots & \dots & 0 & 1 \\ -a_{i,\ell_i} & 0 & \dots & \dots & 0 \end{bmatrix}$$

while the off-diagonal blocks do not have any specific structure. The measurement matrix H is of the form $H = \left[H_1^T,\ H_2^T\right]^T$ where H_1 is a $q \times m_x$ matrix of the following form

$$H_1 = \text{block diag}\left\{[1\ 0\ \cdots\ 0],\ 1 \times \ell_i,\ i = 1, \dots, q\right\} \tag{3.9}$$

H_2 does not have any particular structure and is not relevant. Note that the characteristic polynomial of F, is given by $f(z) = \prod_{i=1}^{q} \left(z^{\ell_i} + a_{i,1} z^{\ell_i - 1} + \ldots + a_{i,\ell_i}\right)$.

If the Hypercuboidal filter is used, then Theorem 3.1 can be extended to the case of vector measurements is as follows.

Theorem 3.4. *It is possible to stabilize* (3.4) *in the mean-squared sense with an* $(R, \beta)-$*anytime code provided*

$$R > R_{v,n} = \frac{1}{n} \sum_{i=1}^{q} \max\left\{0, \log \sum_{j=1}^{\ell_i} |a_{i,j}|\right\}, \quad \beta > \beta_{v,n} = \frac{2}{n} \log_2 \rho\left(\overline{F}\right) \tag{3.10a}$$

Proof. See Appendix 3.9.2 □

The thresholds if one uses an Ellipsoidal filter are given as follows.

Theorem 3.5. *It is possible to stabilize* (3.4) *in the mean-squared sense with an*

$(R, \beta)-$anytime code provided

$$R > R_{ve,n} = \frac{1}{n}\sum_{i=1}^{q} \max\left\{0, \log\left[\sqrt{m_x}\sum_{j=1}^{\ell_i}|a_{i,j}|\theta^{j-1}\right]\right\}, \quad \beta > \beta_{ve,n} = \frac{2}{n}\log_2\rho\left(F\right) \tag{3.11a}$$

where $\theta = \sqrt{\frac{m_x}{m_x-1}}$ □

We skip the proof for Theorem 3.5 since it is very similar to that of Theorem 3.4.

3.7 Discussion — Asymptotics and the Stabilizable Region

The sufficient conditions derived above are non-asymptotic in the sense that measurements are encoded every time step. Alternately, one can encode the measurements every, say, ℓ time steps, and consider the asymptotic rate and exponent needed as ℓ grows. This is often the form in which such sufficient conditions appear in the literature [66, 71, 86]. Even though the sufficient conditions in Sections 3.5 and 3.6 are non-asymptotic, note that they depend only on the system matrices F, H and not on the noise distribution. In order to compare our results with those in the literature, we examine the sufficient conditions in the asymptotic limit of large ℓ.

3.7.1 The Limiting Case

Note that encoding once every ℓ measurements amounts to working with the system matrix F^ℓ. So, one can calculate this limiting rate and exponent by writing the eigenvalues of F, $\{\lambda_i\}_{i=1}^m$, as $\lambda_i = \mu_i^n$ and letting n scale. The following asymptotic result allows us to compare the sufficient conditions above with those in the literature (e.g., [66, 71, 86]).

Theorem 3.6 (The Limiting Case)**.** *Write the eigenvalues of F, $\{\lambda_i\}_{i=1}^{m_x}$, in the form $\lambda_i = \mu_i^n$. Letting n scale, R_n, $R_{v,n}$, $R_{e,n}$, $R_{ev,n}$ converge to R^*, and β_n, $\beta_{v,n}$, $\beta_{e,n}$,*

$\beta_{ev,n}$ converge to β^*, where

$$R^* = \sum_{i:|\mu_i|>1} \log_2 |\mu_i|, \quad \beta^* = 2 \log_2 \max_i |\mu_i| \tag{3.12}$$

Proof. See Appendix 3.9.5. □

For stabilizing plants over deterministic rate-limited channels, [71] showed that a rate $R > R^*$, where R^* is as in (3.12), is necessary and sufficient. So, asymptotically the sufficient condition for the rate R in Theorem 3.1 is tight. But it is not clear if one do with an exponent smaller than $\beta^* = 2 \log_2 \max_i |\mu_i|$ asymptotically when there is no feedback. Though the above limiting case allows one to obtain a tight and an intuitively pleasing characterization of the rate and exponent needed, it should be noted that this may not be operationally practical. For, if one encodes the measurements every ℓ time steps, even though Theorem 3.6 guarantees stability, the performance of the closed-loop system (the LQR cost, say) may be unacceptably large because of the delay we incur. This is what motivated us to present the sufficient condition in the form that we did above.

3.7.2 A Comment on the Trade-Off Between Rate and Exponent

Once a set of rate-exponent pairs (R, β) that can stabilize a plant is available, one would want to identify the pair that optimizes a given cost function. Higher rates provide finer resolution of the measurements while larger exponents ensure that the controller's estimate of the plant does not drift away; however, we cannot have both. One can either coarsely quantize the measurements and protect the bits heavily or quantize them moderately finely and not protect the bits as much. One can easily cook up examples using an LQR cost function with the balance going either way. Studying this trade-off is integral to making the results practically applicable.

3.7.3 Stabilizable Region

Using the asymptotic sufficient condition in Theorem 3.6 and the thresholds on rate and exponent that we will derive in the next Chapter (see Theorem 4.8), we can discuss the range of the eigenvalues of F, i.e., $\{|\mu_i|\}_{i=1}^{m_x}$, for which the η^{th} moment of x_t in (3.4) can be stabilized over some common channels. Since we are interested in the asymptotics, we assume the same limiting case as in Section 3.7.1. Firstly, consider the scalar case, i.e., $m_x = 1$ and let the eigenvalue be μ. An anytime reliable code with rate R and exponent β can stabilize the process in (3.4) for all μ such that

$$\log_2 |\mu| < \min \left\{ R, \frac{\beta}{\eta} \right\}$$

So, a scalar unstable linear process in (3.4) can be stabilized over a MBIOS channel with Bhattacharya parameter ζ provided

$$\log_2 |\mu| < \log_2 |\mu_{\max}| = \sup_{R<C, \beta<E_\zeta(R)} \min \left\{ R, \frac{\beta}{\eta} \right\} \tag{3.13}$$

The stabilizable region as implied by the threshold in [86] is given by

$$\log_2 |\mu| < \log_2 |\mu_{\max}| = \sup_{R<C, \beta<E_r(R)} \min \left\{ R, \frac{\beta}{\eta} \right\}$$

For $\eta = 2$, the stabilizable region for the BEC and BSC is shown in Figure 3.5 where $|\mu_{max}|$ is plotted against the channel parameter. Consider a vector-valued process with unstable eigenvalues $\{|\mu_i|\}_{i=1}^{m}$. Such a process can be stabilized by a rate R and exponent β anytime reliable code provided $R > \sum_{i=1}^{m} \log |\mu_i|$ and $\beta > \log (\max_i |\mu_i|)$. So, given a channel with Bhattacharya parameter ζ for which the rate exponent curve $(R, E_\zeta(R))$ is achievable, the region of unstable eigenvalues that can be stabilized is given by $\{\mu \in \mathbb{R}^m, \mid \exists R < C \ni \sum_{i=1}^{m} \log |\mu_i| < R \text{ and } \log (\max_i |\mu_i|) < E_\zeta(R)\}$, where C is the Shannon capacity of the channel. For example, let $m = 2$ and $\eta = 2$. Figure 3.6a shows the region of $(|\mu_1|, |\mu_2|)$ that can be stabilized over three different channels, a binary symmetric channel with bit flip probability 0.1 and binary erasure

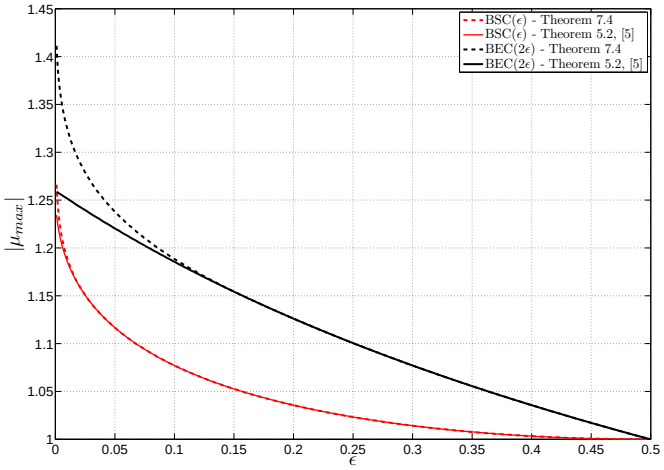
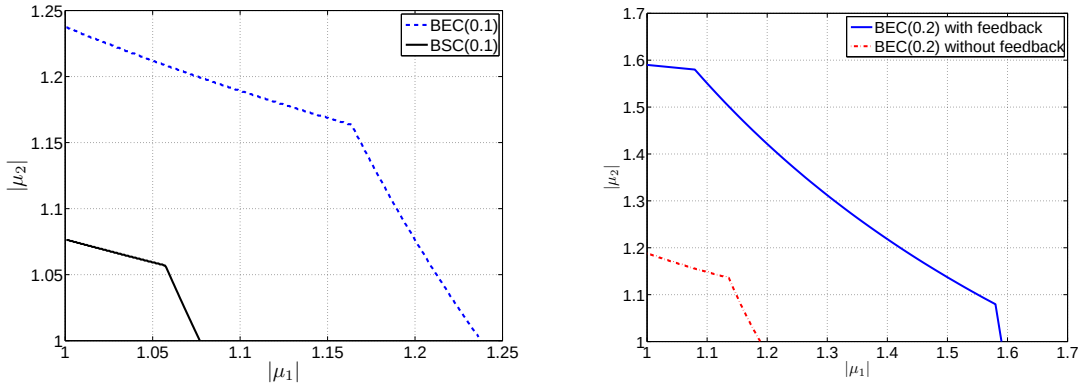


Figure 3.5: Comparing the stabilizable regions of BSC and BEC using linear codes

channels with erasure probabilities 0.1 and 0.2, respectively.

We will now compare these results with the case when there is perfect feedback of the channel outputs at the observer/encoder. [87] considered a priority queuing method for stabilizing vector-valued unstable processes over channels with perfect feedback. Bits from different unstable subsystems are placed in a FIFO queue. Bits are given preference in decreasing order of the size of the eigenvalue of the corresponding subsystem. So, bits coming from a subsystem with a larger eigenvalue are given preference over those from a subsystem with a smaller eigenvalue. A bit is removed from the queue once it is received correctly. Since the feedback anytime capacity of a binary erasure channel is known [85], one can use Theorem 6.1 in [87] to derive the region of eigenvalues that can be stabilized by such a scheme. In Fig. 3.6b, we compare the region of $(|\mu_1|, |\mu_2|)$ that can be stabilized with and without feedback over a binary erasure channel with erasure probability 0.2. As one would expect, the region is much larger when there is feedback. Note that the stabilizable regions in Fig. 3.6 are only achievable and not necessarily tight.



(a) Each curve represents the outer boundary of the stabilizable region.

(b) Stabilizable region with and without feedback

Figure 3.6: Comparing the stabilizable region of different channels

3.8 Summary

We presented various non-asymptotic and hence operationally more meaningful sufficients conditions for stabilizing unstable linear processes over a noisy channel using an $(R, \beta)-$anytime reliable code. Even though the results were non-asymptotic in nature, the thresholds depend only on the properties of the state-space matrix F. The sufficient conditions presented here and in [86, 87] are predicated on the existence of $(R, \beta)-$anytime reliable codes. This is the subject of the next Chapter where we present, for the first time, an explicit ensemble of linear anytime reliable codes.

3.9 Appendices

3.9.1 Proof of Theorem 3.1

The analysis will proceed in two steps. We will first determine a sufficient condition on the number of bits per measurement, nR, that are required to track (3.4) when these bits are available error free. We will then determine the anytime exponent $n\beta$ needed in decoding these source bits when they are communicated over a noisy channel.

Let $\Delta_{t|\tau} \triangleq x_{max,t|\tau} - x_{min,t|\tau}$ be the uncertainty in x_t using $\{b_{\tau'}\}_{\tau' \leq \tau}$, i.e., quantized measurements up to time τ. For convenience, let $\Delta_t \equiv \Delta_{t|t-1}$. Then, the time update is given by the following lemma.

Lemma 3.7 (Time Update)**.** *The time update relating Δ_{t+1} and $\Delta_{t|t}$ is given by* $\Delta_{t+1} = \overline{F}\Delta_{t|t} + W\mathbf{1}_{m_x}$

Proof. From the system dynamics in (3.4), the following is immediate

$$\begin{aligned}
\Delta_{t+1}^{(i)} &= W + \max\left\{\left|\pm a_i \Delta_{t|t}^{(1)} + \Delta_{t|t}^{(i+1)}\right|, \left|\Delta_{t|t}^{(i+1)}\right|, \left|a_i \Delta_{t|t}^{(1)}\right|\right\} \\
&= |a_i|\Delta_{t|t}^{(1)} + \Delta_{t|t}^{(i+1)} + W, \;\; i \leq m-1 \\
\Delta_{t+1}^{(m)} &= |a_m|\Delta_{t|t}^{(1)} + W
\end{aligned}$$

In short, the above equations amount to $\Delta_{t+1} = \overline{F}\Delta_{t|t} + W\mathbf{1}_{m_x}$. □

Towards the measurement update, the observer simply quantizes the measurements y_t according to a $2^{nR}-$regular lattice quantizer with bin width δ, i.e., the quantizer is defined by $Q : \mathbb{R} \mapsto \{0, 1, \ldots, 2^{nR} - 1\}$, where $Q(x) = \lfloor \frac{x}{\delta} \rfloor \bmod 2^{nR}$. In order for this to work, we need $\delta 2^{nR} \geq \Delta_t^{(1)}$ for any time t. Assuming that the rate, R, is large enough, we will first find the steady state value of the recursion for Δ_t, which we then use to determine R. At each time t, the observer can communicate the measurement y_t to within an uncertainty of δ, i.e., the estimator knows that the measurement lies in an interval of width δ. Adding to this the effect of the observation noise, $-\frac{V}{2} \leq v_t \leq \frac{V}{2}$, the estimator knows $x_t^{(1)}$ to within an uncertainty of

$\Delta_{t|t}^{(1)} = \delta + V$. Note that $\Delta_{t|t}^{(i)} = \Delta_t^{(i)}$ for $i \neq 1$. Combining this observation with Lemma 3.7, it is straightforward to see that Δ_t converges, to say Δ_{tu}, in exactly m_x time steps, i.e., $\Delta_t = \Delta_{tu}$ for all $t \geq m_x$. The subscript 'tu' in Δ_{tu} denotes 'time update'. The following result is now immediate.

Lemma 3.8 (Steady State value of Δ_t)**.** $\Delta_{tu} = (\delta + V)L_u a + WL_u \mathbf{1}_{m_x}$, *where* $a = [|a_1|, \ldots, |a_m|]^T$ *and* $L_u = [\ell_{ij}]_{1 \leq i,j \leq m}$ *with* $\ell_{ij} = \mathbb{I}_{i \leq j}$.

Now, we need to go back and calculate R. So we just need

$$\delta 2^{nR} \geq \max\left\{\Delta_0^{(1)}, \Delta_1^{(1)}, \ldots, \Delta_{m_x}^{(1)}\right\}$$

Further, a simple calculation gives

$$\lim_{\delta \to \infty} \frac{\Delta_i^{(1)}}{\delta} = |a_1| + \ldots + |a_i|$$

The minimum rate is thus given by $\frac{1}{n}\log_2 \sum_{i=1}^{m} |a_i|$ and this completes the proof Theorem 3.1.

3.9.2 Proof of Theorem 3.4

The proof is very similar to that of Theorem 3.1. The observations are quantized as follows. At any time, for $1 \leq i \leq q$, the i^{th} component of the measurement vector is quantized using a $2^{nR_i}-$regular lattice quantizer with bin width δ_i. The remaining components of the measurement vector are ignored. The overall rate, R, is then given by $R = R_1 + R_2 \ldots + R_q$. The time update again is given by $\Delta_{t+1} = \overline{F}\Delta_{t|t} + W\mathbf{1}_{m_x}$. The limiting values of $\{R_i\}_{i=1}^{q}$ are obtained by letting $\delta_1 \to \infty$ and $\frac{\delta_i}{\delta_{i+1}} \to \infty$. An argument similar to the one in the previous section gives the following threshold, $R_i \geq \frac{1}{n}\max\{0, \log(|a_{i,1}| + |a_{i,2}| + \ldots + |a_{i,\ell_i}|)\}$.

3.9.3 The Minimum-Volume Ellipsoid

Lemma 3.9 (Theorem 6.1 [38]). *The minimum-volume ellipsoid $\mathcal{E}(\hat{P}, c)$ covering*

$$\left\{x \in \mathbb{R}^m | x \in \mathcal{E}(P, 0), \gamma\sqrt{h^T P h} \leq \langle h, x\rangle \leq \delta\sqrt{h^T P h}\right\}$$

where $|\delta| \geq |\gamma|$, is given by

$$\hat{P} = bP - (b-a)\frac{Phh^T P}{h^T P h}, \;\; c = \xi\frac{Ph}{\sqrt{h^T P h}} \tag{3.14}$$

where

1. *If $\gamma\delta < -\frac{1}{m}$, then $\xi = 0$, $a = b = 1$*

2. *If $\gamma + \delta = 0$ and $\gamma\delta > -\frac{1}{m}$, then*

$$\xi = 0, \;\; a = m\delta^2, \;\; b = \frac{m(1-\delta^2)}{m-1}$$

3. *If $\gamma + \delta \neq 0$ and $\gamma\delta > -\frac{1}{m}$, then*

$$\xi = \frac{m(\gamma+\delta)^2 + 2(1+\gamma\delta) - \sqrt{D}}{2(m+1)(\gamma+\delta)}$$
$$a = m(\xi-\gamma)(\delta-\xi), \;\; b = \frac{a - a\gamma^2}{a - (\xi-\gamma)^2}$$
$$\text{where } D = m^2(\delta^2-\gamma^2)^2 + 4(1-\gamma^2)(1-\delta^2)$$

If $|\delta| < |\gamma|$, change x to $-x$ and apply the above result. And it is easy to verify that $\hat{P}$ is indeed positive semidefinite. Also, a quick calculation shows that $\gamma \leq \xi \leq \delta$. This confirms the intuition that the center of the minimum-volume ellipsoid lies within the slab.

3.9.4 Proof of Theorem 3.3

The proof is in the same spirit as that of Theorem 3.1. We will first determine a sufficient condition on the number of bits per measurement, nR, that are required to track (3.4) when these bits are available error free. We will then determine the anytime exponent $n\beta$ needed in decoding these source bits when they are communicated over a noisy channel.

Consider the time update in (3.7a). Let P_t^{ij} denote the $(i,j)^{th}$ element of P_t, then the time update implies

$$P_{t+1}^{ii} = (1+\epsilon')\left(a_i^2 P_{t|t}^{11} + P_{t|t}^{i+1,i+1} - a_i P_{t|t}^{1,i+1} - a_i P_{t|t}^{i+1,1}\right) + \frac{W^2}{4\epsilon'},\ 1 \le i \le m_x - 1 \tag{3.15a}$$

$$P_{t+1}^{m_x,m_x} = (1+\epsilon')a_{m_x}^2 P_{t|t}^{11} + \frac{W^2}{4\epsilon'} \tag{3.15b}$$

Since the matrix $P_{t|t}$ is positive semidefinite, we have $P_{t|t}^{1,i+1} = P_{t|t}^{i+1,1}$ and $\left(P_{t|t}^{1,i+1}\right)^2 \le P_{t|t}^{11} P_{t|t}^{i+1,i+1}$. Using this in (3.15a), for $1 \le i \le m_x - 1$, we get

$$P_{t+1}^{ii} \le (1+\epsilon')\left(|a_i|\sqrt{P_{t|t}^{11}} + \sqrt{P_{t|t}^{i+1,i+1}}\right)^2 + \frac{W^2}{4\epsilon'} \tag{3.16}$$

This prompts us to bound the recursion (3.7) by bounding the diagonal elements of P_t. Now, considering the measurement update (3.7b), it is easy to see that

$$P_{t|t}^{11} = a_t P_t^{11} \tag{3.17a}$$

$$a_t P_t^{ii} \le P_{t|t}^{ii} \le b_t P_t^{ii} \tag{3.17b}$$

We will first show that $b_t \le \frac{m_x}{m_x - 1}$.

Lemma 3.10. $b_t \le \frac{m_x}{m_x-1}$

Proof. To prove this, consider the setup of Lemma 3.9 and suppose $|\delta| \ge |\gamma|$. Then,

in cases 1) and 2), it is clear that $b \leq \frac{m}{m-1}$ since $|\delta|, |\gamma| \leq 1$. In case 3), we have

$$b = \frac{1-\gamma^2}{1-(\xi-\gamma)^2/a} = \frac{1-\gamma^2}{1-\frac{\xi-\gamma}{m(\delta-\xi)}} \leq \frac{1}{1-\frac{\xi-\gamma}{m(\delta-\xi)}}$$

It suffices to show that $\xi - \gamma \leq \delta - \xi$. This easily follows from the formulae in case 3). The proof for the case when $|\delta| \leq |\gamma|$ is obtained by replacing ξ with $-\xi$. □

As in Section 3.9.1, the observer quantizes the measurements y_t according to a $2^{nR}-$regular lattice quantizer with bin width δ. In order for the controller to know y_t to within a resolution of δ, it is not hard to see that one needs $\delta 2^{nR} > 2\sqrt{P_t^{11}} + v$. We begin by assuming that the rate R is large enough to provide the same resolution δ on y_t at each time t. The actual rate required to accomplish this will be calculated determining an asymptotic upper bound on P_t^{11}. So, at time t, the controller knows that y_t to within a resolution δ and hence $x_t^{(1)}$ to within a resolution of $\delta + V$. Suppose $\sqrt{P_t^{11}}\gamma_t \leq x_t^{(1)} \leq \sqrt{P_t^{11}}\delta_t$, where $\sqrt{P_t^{11}}(\delta_t - \gamma_t) \leq \delta + V$. Then using Lemma 3.9 and noting that $\gamma_t \leq \xi_t \leq \delta_t$, we have

$$\begin{aligned} a_t = m_x(\xi_t - \gamma_t)(\delta_t - \xi_t) &\leq \frac{m_x}{4}(\delta_t - \gamma_t)^2 \\ \implies P_t^{11} a_t &\leq \frac{m_x}{4}(\delta + V)^2 \end{aligned}$$

Using this in (3.17a), we get

$$P_{t|t}^{11} = a_t P_t^{11} \leq \frac{m_x}{4}(\delta + V)^2 \tag{3.18}$$

Combining Lemma 3.10 and (3.18), we get

$$\sqrt{P_{t|t}^{11}} \leq \frac{\sqrt{m_x}}{2}(\delta + V) \tag{3.19a}$$

$$\sqrt{P_{t|t}^{ii}} \leq \sqrt{\frac{m_x}{m_x - 1}}\sqrt{P_t^{ii}}, \;\; i \neq 1 \tag{3.19b}$$

In the following lemma, we will develop an upper bound on the diagonal elements

of P_t which will help us determine an upper bound on P_t^{11}.

Lemma 3.11. *Let $\Delta_{e,0} \in \mathbb{R}^{m_x}$ be such that $\Delta_{e,0}^{(i)} = P_0^{ii}$ for $1 \leq i \leq m_x$ and suppose its evolution is governed by*

$$\Delta_{e,t+1} = (1+\epsilon')^{\frac{1}{2}} \overline{F} \Delta_{e,t|t} + \frac{W}{2\sqrt{\epsilon'}} \mathbf{1}_{m_x}$$

$$\Delta_{e,t|t}^{(i)} = \begin{cases} \delta + V & i = 1 \\ \theta \Delta_{e,t}^{(i)} & i \neq 1 \end{cases}$$

where $\theta = \sqrt{\frac{m_x}{m_x - 1}}$. Then $\sqrt{P_t^{ii}} \leq \Delta_{e,t}^{(i)}$ and $\sqrt{P_{t|t}^{ii}} \leq \Delta_{e,t|t}^{(i)}$ for all t and $1 \leq i \leq m_x$.

Proof. The proof follows by combining the observations from (3.15), (3.16), (3.19). □

Note that the recursion for $\Delta_{e,t}$ above is very similar to that for Δ_t in Section 3.9.1. So, the steady state value of $\Delta_{e,t}^{(1)}$ can be determined by a calculation similar to that in Lemma 3.8. The desired threshold for R is obtained by letting $\delta \to \infty$ for a fixed ϵ'. Since ϵ' can be made arbitrarily small, we get the following bound on R

$$R > \frac{1}{n} \log \left[\sqrt{m_x} \sum_{i=1}^{m_x} |a_i| \theta^{i-1} \right]$$

Now, we need to determine the exponent needed to track (3.4) with a bounded mean-squared error. In the absence of any measurements, it is easy to see from (3.7a) that the growth of P_t is determined by the spectral radius of $\sqrt{1+\epsilon'}F$. Since ϵ' can be made arbitrarily small, in order to track (3.4) with a bounded mean-squared error, we need an anytime exponent $n\beta > 2 \log \rho(F)$. This completes the proof.

3.9.5 The Limiting Case

There are several bounds in the Mathematics literature on the roots of a polynomial in terms of the polynomial coefficients, a standard and near-optimal bound being the Fujiwara's bound which we state below.

Lemma 3.12 (Fujiwara's Bound)**.** *Consider the monic polynomial with complex coefficients* $f(z) = z^m + c_1 z^{m-1} + \ldots + c_m$ *and let* $\rho(f)$ *denote the largest root in magnitude. Then*

$$\rho(f) \leq K(f) = 2 \max \left\{ |c_1|, |c_2|^{\frac{1}{2}}, \ldots, |c_{m-1}|^{\frac{1}{m-1}}, \left| \frac{c_m}{2} \right|^{\frac{1}{m}} \right\}$$

We will detail the proof for the case of scalar measurements. The extension to the vector measurements will then suggest itself. Let F is any m_x-dimensional square matrix and $f(z)$ denotes its characteristic polynomial. Then the following bounds hold (for details see [97])

$$\rho(F) \leq \rho(\overline{F}) \leq \frac{\rho(F)}{\sqrt[m]{2} - 1}, \; K(f) \leq 2\rho(\overline{F}) \tag{3.20}$$

By the hypothesis of the lemma, the eigenvalues of F_n are of the form $\{\mu_i^n\}_{i=1}^{m_x}$. To emphasize the fact that F depends on n, we write it as F_n and a_i as $a_{i,n}$. Recall that the characteristic polynomial of F_n is given by $f_n(z) = z^{m_x} + a_{1,n} z^{m_x - 1} + \ldots + a_{m_x,n}$. Let $\mathcal{I}_u \triangleq \{i \mid |\mu_i| \geq 1\}$, then the following is easy to prove

$$\lim_{n \to \infty} \frac{|a_{i,n}|}{\left|a_{|\mathcal{I}_u|,n}\right|} = 0, \; i \neq |\mathcal{I}_u|, \; \lim_{n \to \infty} \frac{1}{n} \log_2 \left|a_{|\mathcal{I}_u|,n}\right| = \sum_{i \in \mathcal{I}_u} \log_2 |\mu_i| \tag{3.21}$$

From (3.21), it is obvious that $\lim_{n\to\infty} R_n = \sum_{i \in \mathcal{I}_u} \log_2 |\mu_i|$. The asymptotics of $R_{e,n}$, $R_{v,n}$ and $R_{ev,n}$ can be similarly derived. Also, from (3.20), it is clear that $\lim_{n\to\infty} \frac{1}{n} \log \rho\left(\overline{F}_n\right) = \lim_{n\to\infty} \frac{1}{n} \log \rho\left(F_n\right)$. The asymptotics of β_n and $\beta_{v,n}$ now follow immediately.

Chapter 4

Error-Correcting Codes for Interactive Communication

4.1 Introduction

We owe our current understanding of *information* and *communication* to the landmark paper [94], where Shannon laid down the theoretical foundations of modern communication systems. Shannon provided the right mathematical framework to understand and study the problem of transmitting information reliably from a sender to a receiver over an unreliable channel. Reliability is measured by the probability of successfully recovering the message selected by the sender.

Prevailing wisdom at the time seemed to suggest that probability of error cannot be reduced to zero without simultaneously decreasing the rate of communication to zero. Shannon disproved this myth by introducing the idea of block coding which was one of the major breakthroughs in [94]. This is motivated by the observation that a channel is unpredictable over a small number of uses but becomes very predictable when used a large number of times. In other words, if a channel introduces errors with probability p, then in n channel uses, it will introduce approximately np errors with a high probability for large enough n. So, any code of length n that can correct np or more errors will guarantee correct recovery of the message with high probability while achieving a positive rate of communication. This gave rise to the idea of block coding.

For example, as depicted in Figure 4.1, a binary message that needs to be encoded is first divided it into blocks of appropriate size, say k, and each block is separately encoded into a larger block of length, say n, by adding redundancy. The optimal decoder selects the message that is most likely given the channel outputs. The resulting probability of decoding error goes to zero with increasing block length n if an only if the rate of communication, $R = k/n$, is smaller than the Shannon capacity, C, of the channel. After sixty years of coding theory, today we have many practical codes that achieve the Shannon limits in several ways.

The salient features of the setup in Figure 4.1 are 1) communication is one-way, 2) the message is available *a priori* at the sender and 3) the receiver needs to wait until it receives all the n channel outputs before it can decode the message, i.e., delay is not a concern. This paradigm has worked and continues to work very well for many practical delay tolerant applications where communication is essentially one way. These include, data transfers over the internet, telecommunications, deep space communication, data storage, etc. The setup above falls short when communication is fundamentally interactive. We will motivate this through a simple example.

4.1.1 An Example of Interactive Communication

Suppose Alice and Bob wish to carry out a protocol/conversation as depicted in Figure 4.2. Let x and y denote the initial inputs to Alice and Bob, respectively. The objective is to not exchange x and y but to execute a protocol. For example, x and y could denote the two halves of a program input and the protocol could be to compute a function $f(x, y)$ jointly in a distributed manner. In general, the protocol could be anything. In the rest of the discussion, we will use the word "protocol" much the same way as the word "message" is used in information theory. The protocol proceeds as follows. Alice sends a bit $a_0(x)$ to Bob and Bob responds with the bit $b_0(y, a_0(x))$. We call this round 0 of the protocol. Similarly in round i, Alice sends $a_i\left(x, a_{i-1}(.), b_{i-1}(.), \ldots, a_0(.), b_0(.)\right)$ while Bob responds with $b_i\left(y, a_{i-1}(.), b_{i-1}(.), \ldots, a_0(.), b_0(.)\right)$. Suppose that the protocol involves K rounds.

The sequence of all messages transmitted by Alice (Bob) is $X = \{a_0(.), \ldots, a_{K-1}(.)\}$ ($Y = \{b_0(.), \ldots, b_{K-1}(.)\}$). One can treat X (Y) as single message communicated by Alice (Bob) in K rounds. But note that X is not available to Alice a priori, likewise with Bob. It is revealed one bit at a time as the protocol unfolds, unlike the case of one-way communication in Figure 4.1. Consider the problem of executing this protocol reliably over bidirectional noisy channels, i.e., bits sent from Alice (Bob) to Bob (Alice) are subject to i.i.d bit flips, say.

A natural approach in such a setup could be to encode each bit individually using a block code, say of length n (i.e., rate $R = 1/n$), before transmitting it over the channel. If the probability of error for this block code is $p_e(n)$, and the probability that the overall protocol is incorrectly executed is $P_e(K)$, then it is easy to see that $P_e(K) \geq p_e(n)$. Now for $P_e(K) \to 0$, we need $p_e(n) \to$ no matter what K is. But $p_e(n) \to 0$ only when $n \to \infty$ in which case the rate of communication approaches 0. We recovered the same dilemma that faced communication engineers before Shannon's work in [94]. It is thus clear that the conventional block error-correcting codes are not appropriate when communication is fundamentally interactive. The problem of controlling unstable processes over noisy channels that we discussed in Chapter 3 is another instance of interactive communication and as we have seen is not amenable to conventional techniques. In order to reliably simulate interactive protocols, one needs an object called a *tree code* [92] which is essentially a causal encoding scheme satisfying an appropriate Hamming-distance-like property. In spite of the fact that tree codes have been identified to be central to interactive communication problems for nearly two decades now, there has been scant practical progress due to lack of any efficient constructions of tree codes. For the first time, we have an explicit ensemble of linear tree codes that are anytime reliable with high probability.

4.2 Outline

We will begin by defining a tree code in Section 4.3 followed by a literature review in Section 4.4. In Section 4.6, we introduce the notion of causal linear codes and provide

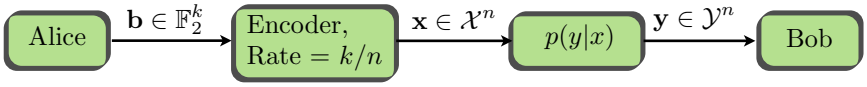


Figure 4.1: A simple schematic to illustrate block coding

a sufficient condition for them to be anytime reliable. The main result appears in ...

4.3 Tree Codes

Let $\mathcal{S}$ be a finite alphabet. If $C = (c_1, \ldots, c_\ell)$ and $C' = (c'_1, \ldots, c'_\ell)$ are words of the same length over $\mathcal{S}$, the Hamming distance between C and C' denoted by $\|C - C'\|$ is the number of positions i in which $c_i \neq c'_i$.

Definition 4.1 (Tree Code [92])**.** *An m-ary tree code over alphabet $\mathcal{S}$, of distance parameter α, is an infinite $m-$ary tree in which every edge of the tree is labeled with a character from the alphabet $\mathcal{S}$ subject to the following condition. Let v_1 and v_2 be any two nodes at some common depth h in the tree. Let $h - d$ be the depth of their least common ancestor. Let $C(v_1)$ and $C(v_2)$ be the concatenation of the letters on the edges leading from the root to v_1 and v_2, respectively. Then $\|C(v_1) - C(v_2)\| \geq \alpha d$.*

The tree defines a causal encoding scheme that at each time τ receives a letter b_τ from an $m-$ary alphabet as input and outputs a letter $c_\tau \in \mathcal{S}$ such that $c_\tau = f_\tau(b_{1:\tau})$, i.e., c_τ is a causal function of the inputs while satisfying the afore-mentioned distance condition. Figure 4.3 depicts a tree code where the input is binary.

4.3.1 Anytime Reliability Under Minimum-Distance Decoding

We will argue concisely why the tree code property is necessary and sufficient for anytime reliability. Recall that an encoder-decoder pair is said to be $(R, \beta)-$anytime reliable over a channel if $P\left(\hat{b}_{\tau+1|t} \neq b_{\tau+1}\right) \leq 2^{-\beta(t-\tau)}$ where the probability is calculated only over the channel realizations. Assume that the channel input alphabet is $\mathcal{S}$

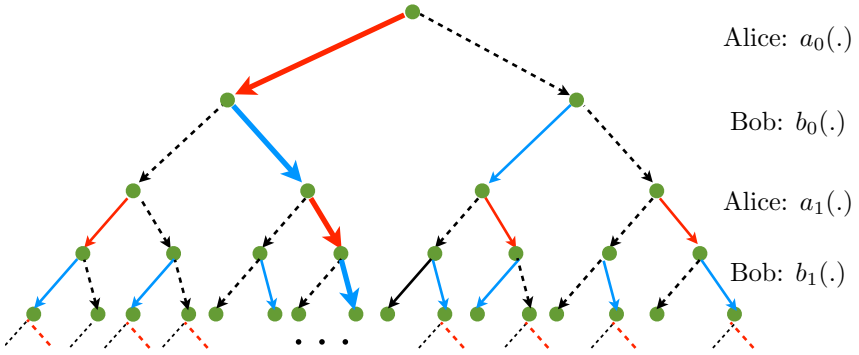


Figure 4.2: The solid edges define the protocol. A realization of the protocol corresponds to a path in the tree. If the protocol is correctly executed, Alice and Bob's messages would correspond to the outlined path

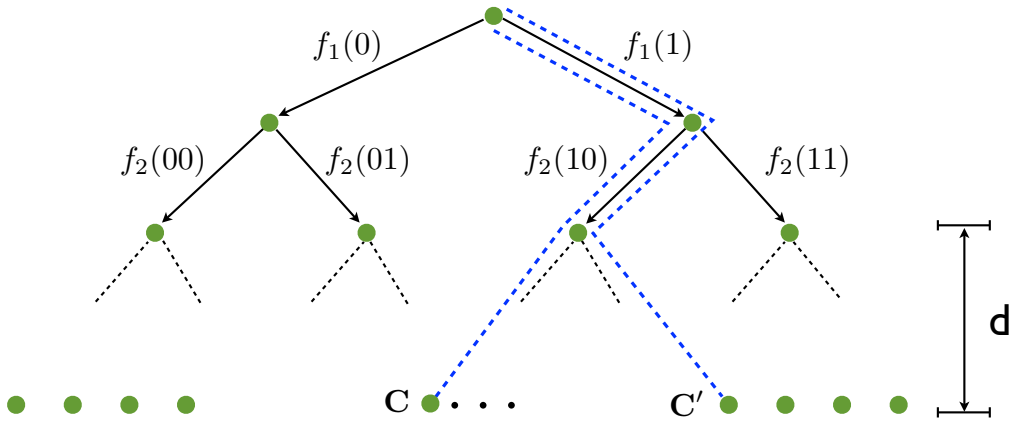


Figure 4.3: One can visualize any causal code on a tree. The distance property is: $\|\mathbf{C} - \mathbf{C}'\|_{\mathcal{H}} \propto d$. This must be true for any two paths with a common root and of equal length in the tree

for simplicity and suppose that the probability that the channel output being different from the input is $\epsilon > 0$. Furthermore suppose that $\epsilon < \alpha/2$ where α is the distance parameter of the tree code and that the channel is memoryless. Then at an arbitrary decoding instant, say t, $\hat{b}_{\tau+1|t} \neq b_{\tau+1}$ implies that the channel must have introduced at least $\alpha(t-\tau)/2$ errors during the time interval $[\tau+1, t]$, the probability of which is at most $2^{-D(\alpha/2,\epsilon)(t-\tau)}$ where $D(\alpha/2, \epsilon)$ is the Kullback-Leibler divergence between Bernoulli($\alpha/2$) and Bernoulli(ϵ). The tree code property is also necessary for anytime reliability. To see this, suppose $C = (c_1, \ldots, c_t)$ is the actual codeword transmitted by the encoder and suppose there exists another codeword $C' = (c'_1, \ldots, c'_t)$ such that $c_1 \neq c'_1$ and $\|C - C'\|$ is sublinear in t, then the probability of confusing between C' and C cannot be smaller than subexponential in t. As a result, the tree code property is necessary and sufficient for anytime reliability.

4.4 Past Work

Early work on the problem of interactive communication over noisy channels appears in [92] where Schulman studied it in the context of distributed computation. Independently in [86], Sahai and Mitter studied it in the context of distributed control. In [92], Schulman introduced a new coding paradigm called *tree codes* and used them to show that one can simulate any interactive protocol between two agents over bidirectional noisy channels with an error probability exponentially small in the length of the protocol, i.e., $P_e(K) \leq 2^{-\Omega(K)}$, while suffering only a constant slowdown. Furthermore, Schulman showed that tree codes exist. This work constitutes an interactive analogue to Shannon's channel coding theorem. Although unlike [94], where Shannon showed that capacity achieving block codes are abundant, Schulman does not show that tree codes exist with high probability. This framework was extended to the case of simulating protocols between a network of agents connected to each other in a graph topology with unreliable links in [79].

In [76,86], it is shown that tree codes under maximum-likelihood (ML) decoding or sequential decoding are anytime reliable. As outlined in Chapter 3, [86] also identifies

that the rate and exponent are crucial to control, unlike [92] where a positive rate is acceptable no matter how small it is.

The problem of simulating protocols over noisy channels has received attention in recent years. In [19], the authors improve upon the algorithm proposed in [92]. If one if willing to tolerate an error probability that is polynomially small in the length of the protocol, then it is possible to come up with explicit code constructions for the case of finite length two-party interactive communication, e.g., [18, 27, 68, 91]. In [34], the authors relax the notion of tree codes to define what they refer to as a *potent tree code* and show how it can be used to simulate any finite length interactive protocol with an error probability that is exponentially small in the length of the protocol. Furthermore, [34] shows that a random construction of a labeled tree produces a potent tree code with high probability.

4.5 Contributions

The explicit code constructions of [18, 27, 68, 91] only guarantee a polynomially small error probability and hence are not applicable in the context of control. It is also not clear if such codes can be used to simulate protocols between more than two agents as is shown possible with tree codes in [79]. We will discuss this in greater detail in Chapter 6. The results in [34] do not apply to control either because potent tree codes are obtained by relaxing the tree code property to allow for large portions of the tree where the Hamming distance property does not hold true and hence the resulting relaxation is not anytime reliable under ML decoding.

Even though the problem of stabilizing unstable processes over noisy channels is an instance of an interactive communication problem, in some ways it places a more stringent requirement on the error-correcting scheme than its counterparts in distributed computation. Furthermore, all the encoding schemes explored thus far are nonlinear in general and do not lend themselves to efficient decoding. A first step in the direction of constructing practical tree codes is to impose linearity. For example, ML decoding of a linear code over an erasure channel just amounts to solving linear

equations which can be accomplished very efficiently. The main contributions of this Chapter are the following

1. We prove that linear anytime reliable codes exist in Section 4.7
2. We demonstrate an explicit ensemble of causal linear codes almost all elements of which are anytime reliable in Section 4.8
3. We present an efficient decoding algorithm for the erasure channel in Section 5.2

We begin by exploring causal linear codes in Section 4.6.

4.6 Linear Anytime Codes

As discussed earlier, a first step towards developing practical encoding and decoding schemes for automatic control is to study the existence of linear codes with anytime reliability. We will begin by defining a causal linear code.

Definition 4.2 (Causal Linear Code)**.** *A causal linear code is a sequence of linear maps* $f_\tau : \mathbb{GF}_2^{k\tau} \mapsto \mathbb{GF}_2^n$ *and hence can be represented as*

$$f_\tau(b_{1:\tau}) = G_{\tau 1}b_1 + G_{\tau 2}b_2 + \ldots + G_{\tau\tau}b_\tau \tag{4.1}$$

where $G_{ij} \in \mathbb{GF}_2^{n\times k}$ □

We denote $c_\tau \triangleq f_\tau(b_{1:\tau})$. Note that a tree code is a more general construction where f_τ need not be linear. Also note that the associated code rate is $R = k/n$. The above encoding is equivalent to using a semi-infinite block lower triangular generator

Table 4.1: Notation for Chapter 4

$\mathbb{H}_{n,R}^t$	$\overline{n}t \times nt$ leading principal minor of $\mathbb{H}_{n,R}$
$\mathcal{C}_t$	$\left\{c \in \{0,1\}^{nt} : \mathbb{H}_{n,R}^t c = 0\right\}$
$\mathcal{C}_{t,d}$	$\{c \in \mathcal{C}_t : c_{\tau<t-d+1} = 0,\ c_{t-d+1} \neq 0\}$
$\|c\|$	Hamming weight of c
$N_{w,d}^t$	$\|\{c \in \mathcal{C}_{t,d} : \|c\| = w\}\|$
$w_{\min,d}^t$	$\text{argmin}_w(N_{w,d}^t \neq 0)$
$P_{t,d}^e$	$P\left(\min\{\tau : \hat{b}_{\tau\|t} \neq b_\tau\} = t-d+1\right)$

matrix $\mathbb{G}_{n,R}$ given by

$$\mathbb{G}_{n,R} = \begin{bmatrix} G_{11} & 0 & \dots & \dots & \dots \\ G_{21} & G_{22} & 0 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ G_{\tau 1} & G_{\tau 2} & \dots & G_{\tau\tau} & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

One can equivalently represent the code with a parity check matrix $\mathbb{H}_{n,R}$, where $\mathbb{G}_{n,R}\mathbb{H}_{n,R} = 0$. The parity check matrix is in general not unique but it is easy to see that one can choose $\mathbb{H}_{n,R}$ to be block lower triangular too.

$$\mathbb{H}_{n,R} = \begin{bmatrix} H_{11} & 0 & \dots & \dots & \dots \\ H_{21} & H_{22} & 0 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ H_{\tau 1} & H_{\tau 2} & \dots & H_{\tau\tau} & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{4.2}$$

where $H_{ij} \in \{0,1\}^{\overline{n}\times n}$ and $\overline{n} = n(1-R)$. In fact, we present all our results in terms of the parity check matrix. Before proceeding further, some of the notation specific to coding is summarized in Table 4.1.

The objective is to study the existence of causal linear codes which are $(R,\beta)-$anytime reliable under maximum-likelihood (ML) decoding. With reference

to Fig. 4.3, this amounts to choosing the branch labels, $f_\tau(b_{1:\tau})$, in such a way that they satisfy the distance property, and also are linear functions of the input, $b_{1:\tau}$. Further, we are interested in characterizing the thresholds on the rate, R, and exponent, β, for which such codes exist. In the interest of clarity, we will begin with a selfcontained discussion of a weak sufficient condition on the distance distribution, $\{N^t_{w,d},\, w^t_{\min,d}\}$ (see Table 4.1), of a causal linear code so that it is anytime reliable under ML decoding. This sufficient condition is an adaptation of the distance property illustrated in Fig. 4.3 to the case of causal linear codes. In Section 4.7, we will demonstrate the existence of causal linear codes that satisfy this sufficient condition. The thresholds thus obtained will be significantly tightened in Section 4.9 by invoking some standard results from random coding literature, e.g., [11, 33].

4.6.1 A Sufficient Condition

Suppose the decoding instant is t and without loss of generality, assume that the all zero codeword is transmitted, i.e., $c_\tau = 0$ for $\tau \leq t$. We are interested in the error event where the earliest error in estimating b_τ happens at $\tau = t - d + 1$, i.e., $\hat{b}_{\tau|t} = 0$ for all $\tau < t-d+1$ and $\hat{b}_{t-d+1|t} \neq 0$. Note that this is equivalent to the ML codeword, $\hat{c}$, satisfying $\hat{c}_{\tau<t-d+1} = 0$ and $\hat{c}_{t-d+1} \neq 0$, and $\mathbb{H}^t_{n,R}$ having full rank so that $\hat{c}$ can be uniquely mapped to a transmitted sequence $\hat{b}$. Then, using a union bound, we have

$$P^e_{t,d} = P\left[\bigcup_{c\in\mathcal{C}_{t,d}} (0 \text{ is decoded as } c)\right] \leq \sum_{c\in\mathcal{C}_{t,d}} P\,(0 \text{ is decoded as } c) \tag{4.3}$$

Consider a *memoryless binary-input output-symmetric* (MBIOS) channel. Let $\mathcal{X}$ and $\mathcal{Z}$ denote the input and output alphabet, respectively. The Bhattacharya parameter, ζ, for such a channel is defined as

$$\zeta = \begin{cases} \int\limits_{z\in\mathcal{Z}} \sqrt{p(z|X=1)p(z|X=0)}dz & \text{if } \mathcal{Z} \text{ is continuous} \\ \sum_{z\in\mathcal{Z}} \sqrt{p(z|X=1)p(z|X=0)} & \text{if } \mathcal{Z} \text{ is discrete valued} \end{cases}$$

Now, it is well known (e.g., [89]) that, under ML decoding

$$P\left(0 \text{ is decoded as } c\right) \leq \zeta^{\|c\|}$$

From (4.3), it follows that $P_{t,d}^{e} \leq \sum_{w_{\min,d}^{t} \leq w \leq nd} N_{w,d}^{t} \zeta^{w}$. If $w_{\min,d}^{t} \geq \alpha nd$ and $N_{w,d}^{t} \leq 2^{\theta w}$ for some $\theta < \log_2(1/\zeta)$, then

$$P_{t,d}^{e} \leq \eta 2^{-\alpha nd(\log_2(1/\zeta)-\theta)} \tag{4.4}$$

where $\eta = (1 - 2^{\log_2(1/\zeta)-\theta})^{-1}$. So, an obvious sufficient condition for $\mathbb{H}_{n,R}$ can be described in terms of $w_{\min,d}^{t}$ and $N_{w,d}^{t}$ as follows. For some $\theta < \log_2(1/\zeta)$, we need

$$w_{\min,d}^{t} \geq \alpha nd \ \forall \ t, \ \ d \geq d_o \tag{4.5a}$$

$$N_{w,d}^{t} \leq 2^{\theta w} \ \forall \ t, \ \ d \geq d_o \tag{4.5b}$$

where d_o is a constant that is independent of d, t. This brings us to the following definition

Definition 4.3 (Anytime distance and Anytime reliability)**.** *We say that a code $\mathbb{H}_{n,R}$ has $(\alpha, \theta, d_o)-$anytime distance, if the following hold*

1. *$\mathbb{H}_{n,R}^{t}$ is full rank for all $t > 0$*
2. *$w_{\min,d}^{t} \geq \alpha nd$, $N_{w,d}^{t} \leq 2^{\theta w}$ for all $t > 0$ and $d \geq d_o$.* □

We require that $\mathbb{H}_{n,R}^{t}$ have full rank so that the mapping from the source bits $b_{1:t}$ to coded bits $c_{1:t}$ is invertible. We summarize the preceding discussion as the following lemma.

Lemma 4.1. *If a code $\mathbb{H}_{n,R}$ has $(\alpha, \theta, d_o)-$anytime distance, then it is $(R, \beta, d_o)-$anytime reliable under ML decoding over a channel with Bhattacharya parameter ζ where $\beta = \alpha\left(\log(1/\zeta) - \theta\right)$* □

4.7 Linear Anytime Codes — Existence

Proving the existence of an anytime reliable causal linear code amounts to proving the existence of a semi-infinite block triangular matrix $\mathbb{H}_{n,R}$ of the form (4.2) with $(\alpha, \theta, d_o)-$anytime distance for some $\alpha > 0$ and $\theta < \log(1/\zeta)$. In order to do so, for each $T > 0$, we will prove by induction the existence of an $\overline{n}T \times nT$ block triangular matrix which we denote by $\mathbb{H}_{n,R,T}$ with $(\alpha, \theta, d_0)-$anytime distance. Using Lemma 4.1, this will give us an $(R, \beta, d_o)-$anytime reliable code over any finite time horizon, i.e., for each $T > 0$, there exists a causal linear code which under ML decoding satisfies

$$P\left(\hat{b}_{t-d+1|t} \neq b_{t-d+1}\right) \leq 2^{-n\beta d}, \ \forall \ d \geq d_0, \ d_0 \leq t \leq T$$

Extension to the limiting case $T \to \infty$ is a technicality and is obtained by a straightforward application of König's lemma (e.g., [58]).

The following lemma proves the existence of a linear anytime reliable code over a finite time horizon.

Lemma 4.2 (Appropriate Weight Distribution)**.** *For each time $T > 0$, rate $R > 0$, $\alpha < H^{-1}(1-R)$ and $\theta > \log(1/(2^{1-R}-1))$, there exists a causal linear code $H(n, k, T)$ that has $(\alpha, \theta, d_o)-$anytime distance, where d_o is a constant independent of d, t and T.*

The proof is by induction and is detailed in the Appendix. Extension to the semi-infinite case is straightforward and we state the result as a theorem.

Theorem 4.3 (Appropriate Weight Distribution)**.** *For rate $R > 0$, $\alpha < H^{-1}(1 - R)$ and $\theta > \log(1/(2^{1-R} - 1))$, there exists a causal linear code $\mathbb{H}(n, R)$ that has $(\alpha, \theta, d_o)-$anytime distance, where d_o is a constant independent of d and t*

We can now use this result to demonstrate an achievable region of rate-exponent pairs for a given channel, i.e., the set of rates R and exponents β such that one can guarantee (R, β) anytime reliability using linear codes. To determine the values of R

that will satisfy (4.4), note that we need

$$\log(1/(2^{1-R}-1)) < \log(1/\zeta) \implies R < 1 - \log(1+\zeta)$$

With this observation, we have the following Corollary.

Corollary 4.4. *For any rate R and exponent β such that*

$$\begin{aligned} &R < 1 - \log(1+\zeta), \quad \textit{and} \\ &\beta < H^{-1}(1-R)\left(\log\left(\frac{1}{\zeta}\right) + \log\left(2^{1-R}-1\right)\right) \end{aligned}$$

there exists a causal linear code that is $(R, \beta, d_0)-$anytime reliable.

Note that for BEC(ϵ), $\zeta = \epsilon$ and for BSC(ϵ), $\zeta = 2\sqrt{\epsilon(1-\epsilon)}$. Theorem 4.3 is equivalent to proving that it is possible to choose labels in Figure 4.3 in such a way that the labels are a linear function of the inputs and the distance property is satisfied. Theorem 4.3 only proves existence of linear tree codes but existence again is not with a high probability. The primary reason for this is the following, one needs

$$P\left(\hat{b}_{t-d+1|t} \neq b_{t-d+1}\right) \leq 2^{-n\beta d} \tag{4.7}$$

to be true for all decoding instants t and all delays d. A natural technique to construct such codes to choose the edge labels at random and insist that (4.7) be true for all t and d. A naïve union bound over both parameters will not even guarantee existence. In fact, in such a random construction, one can show that there will be large portions of the tree where the labels will not satisfy the distance conditions with high probability. This compels one to use an inductive argument. In what follows in Section 4.8, we will remove the need for one of the union bounds by insisting on the code being time invariant. This way, one will only need to guarantee (4.7) for all delays d.

4.8 Linear Time Invariant Codes

Consider causal linear codes with the following Toeplitz structure

$$\mathbb{H}_{n,R}^{TZ} = \begin{bmatrix} H_1 & 0 & \dots & \dots & \dots \\ H_2 & H_1 & 0 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ H_\tau & H_{\tau-1} & \dots & H_1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

The superscript TZ in $\mathbb{H}_{n,R}^{TZ}$ denotes 'Toeplitz'. $\mathbb{H}_{n,R}^{TZ}$ is obtained from $\mathbb{H}_{n,R}$ in (4.2) by setting $H_{ij} = H_{i-j+1}$ for $i \geq j$. Due to the Toeplitz structure, we have the following invariance, $w_{\min,d}^{t} = w_{\min,d}^{t'}$ and $N_{w,d}^{t} = N_{w,d}^{t'}$ for all $d \leq \min(t, t')$. The code $\mathbb{H}_{n,R}^{TZ}$ will be referred to as a time-invariant code. The notion of time invariance is analogous to the convolutional structure used to show the existence of infinite tree codes in [92]. This time invariance allows one to prove that such codes which are anytime reliable are abundant.

Definition 4.4 (The ensemble $\mathbb{TZ}_p$)**.** *The ensemble $\mathbb{TZ}_p$ of time-invariant codes, $\mathbb{H}_{n,R}^{TZ}$, is obtained as follows, H_1 is any fixed full rank binary matrix and for $\tau \geq 2$, the entries of H_τ are chosen i.i.d according to Bernoulli(p), i.e., each entry is 1 with probability p and 0 otherwise.* □

For the ensemble $\mathbb{TZ}_p$, we have the following result

Theorem 4.5 (Abundance of time-invariant codes)**.** *Let $\overline{p} = \min\{p, 1-p\}$. Then, for each $R > 0$ and*

$$\alpha < H^{-1}(1 - R\log(1/(1-\overline{p}))),\ \ \theta > -\log\left[(1-\overline{p})^{-(1-R)} - 1\right],\ \textit{we have}$$
$$P\left(\mathbb{H}_{n,R}^{TZ}\ \textit{has}\ (\alpha, \theta, d_o) - \textit{anytime distance}\right) \geq 1 - 2^{-\Omega(nd_o)}$$

Proof. See Appendix 4.11.1 □

The thresholds on the anytime distance appearing in Theorem 4.5 are same as

those appearing in Theorem 4.3. Hence the associated region of achievable rate-exponent pairs is the same as in Corollary 4.4. The only difference is that Theorem 4.5 refers to the Toeplitz ensemble. We will state this as a separate result as follows.

Corollary 4.6. *For any rate R and exponent β such that*

$$R < 1 - \log(1+\zeta), \quad \textit{and}$$
$$\beta < H^{-1}(1-R)\left(\log\left(\frac{1}{\zeta}\right) + \log\left(2^{1-R}-1\right)\right)$$

if $\mathbb{H}_{n,R}^{TZ}$ is chosen from $\mathbb{TZ}_{\frac{1}{2}}$, then

$$P\left(\mathbb{H}_{n,R}^{TZ} \textit{ is } (R,\beta,d_o) - \textit{anytime reliable}\right) \geq 1 - 2^{-\Omega(nd_o)}$$

□

The constant in the exponent $\Omega(nd_o)$ in Corollary 4.4 can be computed explicitly and it decreases to zero if either the rate or the exponent approach their respective thresholds. Further note that almost every code in the ensemble is (R,β)-anytime reliable after a large enough initial delay d_o. In other words, a code in the ensemble is not anytime reliable implies that there is no finite delay d_0 beyond which (4.7) holds, the probability of which is 0 by Corollary 4.6.

The Role of the constant d_0 - For the purpose of stabilizing unstable plants over noisy channels, it is sufficient to guarantee exponentially decaying error probability for delays larger than any finite constant. This motivated the constant d_0 when we defined the notion of anytime reliability in Definition 3.1. The role of d_0 in simulating general protocols between two or more agents is more tricky. If the channels connecting the agents are erasure links, the only effect d_0 will have is to slowdown the protocol further but only by a constant factor. In other words, $(R,\beta,d_0)-$anytime reliable codes can be used to simulate protocols between a network of agents connected to each other with erasure links. But when the channels are not erasure links, it is not clear if $(R,\beta,d_0)-$anytime reliable codes can be used to simulate general protocols when $d_0 > 1$.

The thresholds in Corollary 4.4 have been obtained by using a simple union bound for bounding the error probability in (4.3). As one would expect, these thresholds can be improved by doing a more careful analysis. It turns out that the ensemble of random causal linear codes bears close resemblance to random linear block codes. This allows one to borrow results from the random coding literature to tighten the thresholds.

4.9 Improving the Thresholds

We will examine the Toeplitz ensemble more closely and show that its delay-dependent distance distribution is bounded above by that of the random binary linear code ensemble, which we will define shortly. This will enable us to significantly improve the rate, exponent thresholds of Section 4.7 that were obtained using a simple union bound.

4.9.1 A Brief Recap of Random Coding

For an arbitrary discrete memoryless channel, recall the following familiar definition of the random coding exponent, $E_r(R)$, from [33][1]

$$E_r(R) = \max_{0\leq\rho\leq 1} \max_{\mathbf{Q}} \left[E_o\left(\rho, \mathbf{Q}\right) - \rho R\right], \text{ where} \tag{4.9a}$$

$$E_o\left(\rho, \mathbf{Q}\right) = -\log_2 \sum_{z\in\mathcal{Z}} \left[\sum_{x\in\mathcal{X}} Q(x)p(z|X=x)^{\frac{1}{1+\rho}}\right]^{1+\rho} \tag{4.9b}$$

In (4.9b), $Q(.)$ denotes a distribution on the channel input alphabet. The ensemble of random binary linear codes with block length N and rate $R = \frac{K}{N}$ is obtained by choosing an $(N-K)\times N$ binary parity check matrix H, i.e., $H \in GF_2^{(N-K)\times N}$, each of whose entries is chosen i.i.d Bernoulli$\left(\frac{1}{2}\right)$. For such an ensemble, any nonzero binary word $c \in GF_2^N$ is a codeword with probability $2^{-N(1-R)}$. For a given block code, let $w_{\min}$ denote the minimum-distance and N_w the number of codewords with Hamming

[1]We use base-2 instead of the natural logarithm

weight w. A quick calculation shows that $\mathbb{E}N_w = \binom{N}{w}2^{-N(1-R)}$ and that $w_{\min}$ grows like $H^{-1}(1-R)N$ with a high probability. A *typical* code in this ensemble is defined to be one that has $w_{\min} \approx H^{-1}(1-R)N$ and $N_w \approx \binom{N}{w}2^{-N(1-R)}$. A simple Markov inequality shows that the probability that a code from this ensemble is *atypical* is at most $2^{-\Omega(N)}$. For the typical code over BSC(ϵ), the block error probability decays as $2^{-NE_{BSC}(R)}$ where the exponent E_{BSC} has been characterized in [11]. As has been noted in [11], these calculations can be easily extended to a wider class of channels. In particular, the class of MBIOS channels admits a particularly clean characterization. We present the following generalization of the result in [11] without proof.

Lemma 4.7. *Consider a linear code with block length N, rate R and distance distribution $\{N_w\}_{w=1}^{N}$ such that*

1. *$N_w = 0$ if $w \leq H^{-1}(1-R-\delta)$*
2. *$N_w \leq 2^{-N(1-R-\delta+o(1))}\binom{m}{w}$*

for some $\delta > 0$. Let the channel be a MBIOS channel with Bhattacharya parameter ζ. Then the block error probability, P_e, under ML decoding is bounded as

$$P_e \leq 2^{-N\left(E_\zeta(R)-\delta'\right)} \tag{4.10}$$

where

$$E_\zeta(R) = \begin{cases} H^{-1}(1-R)\log\frac{1}{\zeta} & , \ 0 \leq R \leq 1 - H\left(\frac{\zeta}{1+\zeta}\right) \\ E_r(R) & , \ 1 - H\left(\frac{\zeta}{1+\zeta}\right) \leq R \leq C \end{cases} \tag{4.11}$$

and $\delta' \to 0$ as $\delta \to 0$.

Proof. The proof is a straightforward generalization of the result in [11]. □

4.9.2 The Toeplitz Ensemble

In the causal case, fix an arbitrary decoding instant t and consider the event that the earliest error happens at a delay d. As seen before, the associated error probability

depends on the relevant codebook $\mathcal{C}_{t,d}$ and its distance distribution $\{N_{w,d}^t\}_{w=1}^{nd}$. Recall from Table 4.1 that

$$\mathcal{C}_{t,d} \triangleq \{c \in \mathcal{C}_t : c_{\tau<t-d+1} = 0,\ c_{t-d+1} \neq 0\}$$

Due to the Toeplitz structure, we have $\mathcal{C}_{t,d} = \mathcal{C}_{d,d}$. So, we drop the subscript t in $N_{w,d}^t$ and write it as $N_{w,d}$. Note that $\mathcal{C}_{d,d}$ is determined by the matrix $\mathbb{H}_{n,R}^d$. Let c be a given nd-dimensional binary word, i.e., $c \in GF_2^{nd}$, and write $c = \left[c_1^T, c_2^T, \ldots, c_d^T\right]^T$, where $c_\tau \in GF_2^n$ notionally corresponds to the n encoder output bits during the τ^{th} time slot. Suppose $c_1 \neq 0$, then it is easy to see that

$$P\left(\mathbb{H}_{n,R}^d c = 0\right) = 2^{-\overline{n}d}$$

Recall that $\overline{n} = n(1-R)$.

Now observe that $\mathbb{E}N_{w,d} \leq \binom{nd}{w} 2^{-\overline{n}d}$. This is same as the average weight distribution of the random binary linear code with a block length nd and rate R. So, applying Lemma 4.7, we get the following result.

Theorem 4.8. *For each rate $R < C$ and exponent $\beta < E_\zeta(R)$, if $\mathbb{H}_{n,R}^{TZ}$ is chosen from $\mathbb{TZ}_{\frac{1}{2}}$, then*

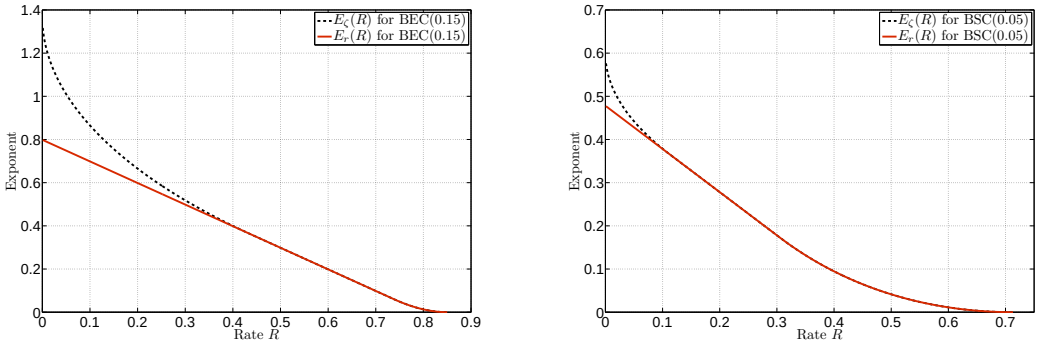
$$P\left(\mathbb{H}_{n,R}^{TZ} \textit{ is } (R, \beta, d_o) - \textit{anytime reliable}\right) \geq 1 - 2^{-\Omega(nd_o)}$$

where C is the Shannon capacity of the channel and

$$E_\zeta(R) = \begin{cases} H^{-1}(1-R)\log\frac{1}{\zeta} & ,\ 0 \leq R \leq 1 - H\left(\frac{\zeta}{1+\zeta}\right) \\ E_r(R) & ,\ 1 - H\left(\frac{\zeta}{1+\zeta}\right) \leq R \leq C \end{cases} \tag{4.12}$$

□

The problem of stabilizing unstable scalar linear systems over noisy channels in the absence of feedback has been considered in [86]. [86] showed the existence of $(R, \beta)-$anytime reliable codes for $R < C$ and $\beta < E_r(R)$. The code is not linear in



(a) Binary Erasure Channel, $\epsilon = 0.15$ (b) Binary Symmetric Channel, $\epsilon = 0.05$

Figure 4.4: Comparing the thresholds obtained from Theorem 4.8 and Theorem 5.2 in [86]

general and the existence was not with high probability. Theorem 4.8 proves linear anytime reliable codes for exponent, β, up to $E_\zeta(R)$. When $R < 1 - H\left(\frac{\zeta}{1+\zeta}\right)$, $E_\zeta(R) > E_r(R)$. So, Theorem 4.8 marks a significant improvement in the known thresholds for stabilizing unstable processes over noisy channels, as is demonstrated in Figures 4.4 and 3.5.

4.10 Summary

The sufficient conditions on the rate and exponent of anytime reliable codes developed in [86, 87] and Chapter 3 are predicated upon the existence of error-correcting codes that achieve such reliabilities. One needs tree codes in order to achieve anytime reliability over memoryless channels under maximum-likelihood (ML) decoding. Tree codes first appeared independently in the work of Schulman [79, 92] in a different context of distributed computation. Schulman used tree codes to simmulate interactive protocols between a network of agents and showed that tree codes exist effectively providing an interactive analogue of Shannon channel coding theorem which considered one way communication.

Even though the significance of tree codes in interactive communication problems has been understood for nearly two decades, there have been no practical construc-

tions till date. The existence of tree codes proved in [92] is not with high probability. The codes are also nonlinear in general and do not lend themselves to efficient decoding. In this Chapter, we attempted to bridge this gap in our understanding of tree codes. For the first time, we showed the existence of linear tree codes. Moreover we show that codes drawn from an appropriate time-invariant ensemble are anytime reliable with a high probability. In other words, we prove that codes drawn from an appropriate ensemble of causal linear codes which we call the *Toeplitz ensemble* are $(R, \beta)-$anytime reliable with high probability for rates upto Shannon capacity and exponent up to the expurgated exponent [11]. This significantly improves upon the known rate and exponent pairs for which anytime reliable codes are known to exist. In the next Chapter, we exploit the linearity of the codes to decode them efficiently over erasure channels.

4.11 Appendices

4.11.1 Proof of Theorem 4.5

We will begin with some preliminary observations.

Lemma 4.9 ([51]). *Let V be an $m-$dimensional vector space over $\mathbb{GF}_2$ and define a probability function over V such that, for each $v \in V$, $P(v) = p^{\|v\|}(1-p)^{m-\|v\|}$. If U is an $\ell-$dimensional subspace of V, then*

$$P(U) \leq \max(p, 1-p)^{m-\ell}$$

Proof. Suppose $p \leq 1/2$. The proof for the other case is analogous. Let E be the set of unit vectors, i.e., $E = \{v \in V \,|\, \|v\| = 1\}$. Then there is a subset, E', of E with $m - \ell$ unit vectors such that $V = U \oplus span(E')$ and $U \cap span(E') = \{0\}$. Let $u' \in span(E')$, then

$$P\left(U + u'\right) = \sum_{u \in U} P(u + u') \geq \sum_{u \in U} P(u) \left(\frac{p}{1-p}\right)^{\|u'\|} = P(U) \left(\frac{p}{1-p}\right)^{\|u'\|}$$

Note that for distinct $u'_1, u'_2 \in span(E')$, $(U + u'_1) \cap (U + u'_2) = \emptyset$. Also note that $\|u'\| \leq m - \ell \;\forall\; u' \in span(E')$.

$$1 = P(V) = P\left(\bigcup_{u' \in span(E')} (U + u')\right) \geq \sum_{u' \in span(E')} P(U) \left(\frac{p}{1-p}\right)^{\|u'\|}$$

Observe that there are exactly $\binom{m-\ell}{i}$ vectors in $span(E')$ with Hamming weight i. So, we have

$$1 \geq P(U) \sum_{i=0}^{m-\ell} \binom{m-\ell}{i} \left(\frac{p}{1-p}\right)^{i} = P(U) \left(\frac{1}{1-p}\right)^{m-\ell}$$

This completes the proof. □

Remark 4.1. *The Toeplitz parity check matrix $\mathbb{H}_{n,R}^{TZ}$ is full rank if and only if H_1*

is full rank. This is why we fix H_1 to be a full rank matrix in the definition of the Toeplitz ensemble.

Recall that we choose the entries of H_i to be i.i.d Bernoulli(p) for $i \geq 2$. Also suppose $p \leq 1/2$. The results for $p \geq 1/2$ are obtained by replacing p with $1-p$ in the subsequent analysis. Consider an arbitrary decoding instant, t. Since $w^t_{\min,d} = w^{t'}_{\min,d}$ and $N^t_{w,d} = N^{t'}_{w,d}$ for all t, t', we will drop these superscripts and write $w^t_{\min,d} = w_{\min,d}$ and $N^t_{w,d} = N_{w,d}$. Let $c = [c_1^T, \ldots, c_t^T]^T$, where $c_i \in \{0,1\}^n$, be a fixed binary word such that $c_{\tau<t-d+1} = 0$ and $c_{t-d+1} \neq 0$. Also, let $\mathbb{H}_{n,R}$ be drawn from the ensemble $\mathbb{TZ}_p$ and let $\mathbb{H}^t_{n,R}$ denote the $\overline{n}t \times nt$ principal minor of $\mathbb{H}_{n,R}$. We examine the probability that c is a codeword of $\mathbb{H}^t_{n,R}$, i.e., $P\left(\mathbb{H}^t_{n,R}c = 0\right)$. Now, since $c_{\tau<t-d+1} = 0$, $\mathbb{H}^t_{n,R}c = 0$ is equivalent to

$$\begin{bmatrix} H_1 & 0 & \ldots & \ldots \\ H_2 & H_1 & 0 & \ldots \\ \vdots & \vdots & \ddots & \vdots \\ H_d & H_{d-1} & \ldots & H_1 \end{bmatrix} \begin{bmatrix} c_{t-d+1} \\ c_{t-d+2} \\ \vdots \\ c_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.13}$$

Note that (4.13) can be equivalently written as follows

$$\begin{bmatrix} C_{t-d+1} & 0 & \ldots & \ldots \\ C_{t-d+2} & C_{t-d+1} & 0 & \ldots \\ \vdots & \vdots & \ddots & \vdots \\ C_t & C_{t-1} & \ldots & C_{t-d+1} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.14}$$

where $h_i = \text{vec}(H_i^T)$, i.e., h_i is a $n\overline{n} \times 1$ column obtained by stacking the columns of H_i^T one below the other, and $C_i \in \{0,1\}^{\overline{n} \times n\overline{n}}$ is obtained from c_i as follows.

$$C_i = \begin{bmatrix} c_i^T & 0 & \ldots & \ldots \\ 0 & c_i^T & 0 & \ldots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & c_i^T \end{bmatrix} \tag{4.15}$$

Since H_1 is fixed, we will rewrite (4.14) as

$$\underbrace{\begin{bmatrix} C_{t-d+1} & 0 & \dots & \dots \\ C_{t-d+2} & C_{t-d+1} & 0 & \dots \\ \vdots & \vdots & \ddots & \vdots \\ C_{t-1} & C_{t-2} & \dots & C_{t-d+1} \end{bmatrix}}_{\triangleq C} \underbrace{\begin{bmatrix} h_2 \\ h_3 \\ \vdots \\ h_d \end{bmatrix}}_{\triangleq h} = \begin{bmatrix} C_{t-d+2} \\ C_{t-d+3} \\ \vdots \\ C_t \end{bmatrix} h_1, \;\; C_{t-d+1}h_1 = 0 \quad (4.16)$$

Since $c_{t-d+1} \neq 0$, C_{t-d+1} has full rank $\overline{n}$ and consequently C has full rank $(d-1)\overline{n}$. Since C is an $(d-1)\overline{n} \times (d-1)n\overline{n}$ matrix, its null space has dimension $(d-1)n\overline{n} - (d-1)\overline{n}$. For (4.16) to hold, h must lie in an $(d-1)n\overline{n} - (d-1)\overline{n}$ dimensional flat which is contained in an $(d-1)n\overline{n} - (d-1)\overline{n} + 1$ dimensional subspace. Using Lemma 4.9, we have

$$P(\mathbb{H}^t_{n,R} c = 0) \leq (1-p)^{\overline{n}(d-1)-1} \quad (4.17)$$

$$\begin{aligned} \implies P\left(w_{\min,d} < \alpha nd\right) &\leq (1-p)^{\overline{n}(d-1)-1} \sum_{w' \leq \alpha nd} \binom{nd}{w'} \\ &\leq (1-p)^{\overline{n}(d-1)-1} 2^{ndH(\alpha)} \\ &= \eta 2^{-nd((1-R)\log_2(1/(1-p)) - H(\alpha))} \end{aligned} \quad (4.18)$$

where $\eta = (1-p)^{-\overline{n}-1}$. Similarly,

$$\begin{aligned} P\left(N_{w,d} > 2^{\theta w}\right) &\leq 2^{-\theta w} \mathbb{E} N_{w,d} \\ &\leq \eta 2^{-\theta w} \binom{nd}{w} (1-p)^{\overline{n}d} \\ &\leq \eta 2^{-nd(\theta w/nd - H(w/nd) + (1-R)\log_2(1/(1-p)))} \end{aligned} \quad (4.19)$$

For convenience, define

$$\begin{aligned} \delta_1 &= (1-R)\log_2(1/(1-p)) - H(\alpha) \\ \delta_{2,w} &= \theta \frac{w}{nd} - H\left(\frac{w}{nd}\right) + (1-R)\log_2(1/(1-p)) \end{aligned}$$

We need to choose θ such that $\delta_{2,w} > \delta > 0$ for all $\alpha \leq \frac{w}{nd} \leq 1$. Now, define

$$\theta^* = \max_{x \geq \alpha} \frac{H(x) - (1-R)}{x} \tag{4.20}$$

Then for each $\theta > \theta^*$, there is a $\delta > 0$ such that $\delta_{2,w} > \delta$ for all $\alpha nd \leq w \leq nd$. A simple calculation gives $\theta^* = \log_2\left(\frac{1}{2^{1-R}-1}\right)$. For such a choice of $\theta > \theta^*$, continuing from (4.19), we have

$$P\left(\exists\, \alpha nd \leq w \leq nd \ \ni \ N_{w,d} > 2^{\theta w}\right) \leq nd2^{-nd\delta} \tag{4.21}$$

for some $\delta' > 0$. For some fixed d_o large enough, applying a union bound over $d \geq d_o$ to (4.18) and (4.21), we get

$$P\left(\exists\ d \geq d_o \ \ni \ w_{\min,d} < \alpha nd \text{ or } N_{w,d} > 2^{\theta w}\right) \leq 2^{-\Omega(nd_o)} \tag{4.22}$$

4.11.2 Proof of Theorem 4.3

The proof is by induction. Suppose $\mathbb{H}_{n,R,T-1}$ has $(\alpha, \theta, d_o)-$anytime distance. Construct $\mathbb{H}_{n,R,T}$ as follows.

$$\mathbb{H}_{n,R,T} = \left[\begin{array}{c|ccc} H_{11} & 0 & \dots & \dots \\ \hline H_{21} & & & \\ \vdots & & \mathbb{H}_{n,R,T-1} & \\ H_{T1} & & & \end{array}\right]$$

where H_{11} is chosen to be a full rank matrix and the entries of $H_{j1} \in \{0,1\}^{\overline{n} \times n}$, $j \geq 2$, are drawn according to i.i.d Bernoulli($\frac{1}{2}$). We will show that if $\mathbb{H}_{n,R,T-1}$ has $(\alpha, \theta, d_o)-$anytime distance, then $\mathbb{H}_{n,R,T}$ will is also have $(\alpha, \theta, d_o)-$anytime distance with a probability $1-2^{-\Omega(nd_o)}$. Note that the probability is over the choice of $\{H_{j1}\}_{j=1}^{T}$. Let $\{w^t_{\min,d}, N^t_{w,d}\}_{d \geq d_o, t \leq T}$ be the weight distribution parameters associated to $\mathbb{H}_{n,R,T}$.

Since $\mathbb{H}_{n,R,T-1}$ has $(\alpha, \theta, d_o)-$anytime distance, we have the following

$$w^t_{\min,d} \geq \alpha n d, \ \forall \ d_o \leq d \leq t-1, \ t \geq d_o + 1$$
$$N^t_{w,d} \leq 2^{\theta w}, \ \forall \ w \geq \alpha n d, \ d_o \leq d \leq t-1, \ t \geq d_o + 1$$

Towards proving that $\mathbb{H}_{n,R,T}$ has $(\alpha, \theta, d_o)-$anytime distance, it remains to show the following holds with a positive probability.

$$\text{For } t \geq d_o, \ \ w^t_{\min,t} \geq \alpha n t, \ \ N^t_{w,t} \leq 2^{\theta w}, \ \forall \ w \geq \alpha n t \tag{4.23}$$

Recall the notation from Table 4.1. Let $c \in \{0,1\}^{nt}$ such that $c_{\tau < t-d+1} = 0$ and $c_{t-d+1} \neq 0$, then it is easy to see that $P\left(\mathbb{H}^t_{n,R,T} c = 0\right) = 2^{-n(d-1)}$. The rest of the analysis follows exactly along the lines of the proof of Theorem 4.5 starting from (4.17) with $p = \frac{1}{2}$. This gives the following result

$$P\left(\mathbb{H}_{n,R,T} \text{ is bad} | \mathbb{H}_{n,R,T-1} \text{ is good}\right) =$$
$$P\left(\{w^t_{\min,d}, N^t_{w,d}\} \text{ do not satisfy (4.23)}\right) \leq 1 - 2^{-\Omega(n d_o)}$$

In particular, there exists a choice of $\{H_{j1}\}_{j=1}^T$ such that $\mathbb{H}_{n,R,T}$ has $(\alpha, \theta, d_o)-$anytime distance, whenever $\mathbb{H}_{n,R,T-1}$ has $(\alpha, \theta, d_o)-$anytime distance. For the inductive argument to be complete, one needs to prove that there exists a $\mathbb{H}_{n,R,d_o}$ that has $(\alpha, \theta, d_o)-$anytime distance. This is already covered in the proof of the above inductive step.

Chapter 5

Efficient Decoding Over Erasure Channels

5.1 Introduction

We have seen in Chapter 4 that tree codes under ML decoding are anytime reliable. The complexity of performing ML decoding at any decoding instant t is exponentially large in t which quickly becomes infeasible as t grows. The same was true of Shannon's noisy channel coding Theorem in [94] which required either typical set decoding or maximum-likelihood decoding both of which required computation exponential in the block length. An early and a very successful response to this problem was the technique of sequential decoding introduced in the work of [28, 47, 48]. While the ML decoder tries to find the most likely path in the coding tree by searching exhaustively, the sequential decoder does so by searching only locally and hence performing far fewer computations. The amount of computation performed by a sequential decoder is stochastic and the average amount of computation per decoding instant is finite if and only if the rate is smaller than the computational cutoff rate denoted by R_0. For a binary erasure channel with erasure probability ϵ, R_0 is $1 - \log(1 + \epsilon)$. In other words the computational savings afforded by the sequential decoder over the ML decoder are meaningful only when the rate is smaller than R_0. In particular, at any decoding instant, the probability that one has to perform L computations decays as $L^{-\gamma}$ and $\gamma > 1$ iff and only if the rate is smaller than R_0. The authors in [76]

observed that tree codes under sequential decoding are anytime reliable while [92] also proposes a simple albeit suboptimal variant of a sequential decoder. Till date, the sequential decoder is computationally the best known method for decoding tree codes. In this chapter, we will exploit the linearity of the anytime reliable codes discussed in Chapter 4 to propose an efficient ML decoder for the erasure channel. In Section 5.4, we will propose an idea on how to construct efficiently decodable codes for the binary symmetric channel.

5.2 Decoding Over the Binary Erasure Channel

Owing to the simplicity of the erasure channel, it is possible to come up with an efficient way to perform maximum-likelihood decoding at each time step. Consider an arbitrary decoding instant t, let $c = [c_1^T, \ldots, c_t^T]^T$ be the transmitted codeword and let $z = [z_1^T, \ldots, z_t^T]^T$ denote the corresponding channel outputs. Recall that $\mathbb{H}_{n,R}^t$ denotes the $\overline{n}t \times nt$ leading principal minor of $\mathbb{H}_{n,R}$. Let z_e denote the erasures in z and let H_e denote the columns of $\mathbb{H}_{n,R}^t$ that correspond to the positions of the erasures. Also, let $\tilde{z}_e$ denote the unerased entries of z and let $\tilde{H}_e$ denote the columns of $\mathbb{H}_{n,R}^t$ excluding H_e. So, we have the following parity check condition on z_e, $H_e z_e = \tilde{H}_e \tilde{z}_e$. Since $\tilde{z}_e$ is known at the decoder, $s \triangleq \tilde{H}_e \tilde{z}_e$ is known. Maximum-likelihood decoding boils down to solving the linear equation $H_e z_e = s$. Due to the lower triangular nature of H_e, unlike the case of traditional block coding, this equation will typically not have a unique solution, since H_e will typically not have full column rank. This is alright as we are not interested in decoding the entire z_e correctly, we only care about decoding the earlier entries accurately. If $z_e = [z_{e,1}^T,\ z_{e,2}^T]^T$, then $z_{e,1}$ corresponds to the earlier time instants while $z_{e,2}$ corresponds to the latter time instants. The desired reliability requires one to recover $z_{e,1}$ with an exponentially smaller error probability than $z_{e,2}$. Since H_e is lower triangular, we can write $H_e z_e = s$ as

$$\begin{bmatrix} H_{e,11} & 0 \\ H_{e,21} & H_{e,22} \end{bmatrix} \begin{bmatrix} z_{e,1} \\ z_{e,2} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \tag{5.1}$$

Let $H_{e,22}^{\perp}$ denote the orthogonal complement of $H_{e,22}$, i.e., $H_{e,22}^{\perp}H_{e,22} = 0$. Then multiplying both sides of (5.1) with $\mathrm{diag}(I, H_{e,22})$, we get

$$\begin{bmatrix} H_{e,11} \\ H_{e,22}^{\perp}H_{e,21} \end{bmatrix} z_{e,1} = \begin{bmatrix} s_1 \\ H_{e,22}^{\perp}s_2 \end{bmatrix} \tag{5.2}$$

If $[H_{e,11}^T \ (H_{e,22}^{\perp}H_{e,21})^T]^T$ has full column rank, then $z_{e,1}$ can be recovered exactly. The decoding algorithm now suggests itself, i.e., find the smallest possible $H_{e,22}$ such that $[H_{e,11}^T \ (H_{e,22}^{\perp}H_{e,21})^T]^T$ has full rank and it is outlined in Algorithm 3. Note that one

Algorithm 3 Decoder for the BEC

1. Suppose, at time t, the earliest uncorrected error is at a delay d. Identify z_e and H_e as defined above.
2. Starting with $d' = 1, 2, \ldots, d$, partition

   $$z_e = [z_{e,1}^T \ z_{e,2}^T]^T \text{ and } H_e = \begin{bmatrix} H_{e,11} & 0 \\ H_{e,21} & H_{e,22} \end{bmatrix}$$

   where $z_{e,2}$ correspond to the erased positions up to delay d'.
3. Check whether the matrix $\begin{bmatrix} H_{e,11} \\ H_{e,22}^{\perp}H_{e,21} \end{bmatrix}$ has full column rank.
4. If so, solve for $z_{e,1}$ in the system of equations

   $$\begin{bmatrix} H_{e,11} \\ H_{e,22}^{\perp}H_{e,21} \end{bmatrix} z_{e,1} = \begin{bmatrix} s_1 \\ H_{e,22}^{\perp}s_2 \end{bmatrix}$$

5. Increment $t = t + 1$ and continue.

can equivalently describe the decoding algorithm in terms of the generator matrix and it will be very similar to Alg 3.

5.2.1 Encoding and Decoding Complexity

Consider the decoding instant t and suppose that the earliest uncorrected erasure is at time $t - d + 1$. Then steps 2) and 3) in Algorithm 3 can be accomplished by just reducing H_e into the appropriate row echelon form, which has complexity $\mathrm{O}(d^3)$. The

earliest entry in z_e is at time $t-d+1$ implies that it was not corrected at time $t-1$, the probability of which is $P^e_{d-1,t-1} \leq \eta 2^{-n\beta(d-1)}$. Hence, if nothing more had to be done, the average decoding complexity would have been at most $K\sum_{d>0} d^3 2^{-n\beta d}$ which is bounded and is independent of t. In particular, the probability of the decoding complexity being Kd^3 would have been at most $\eta 2^{-n\beta d}$. But, in order to actually solve for $z_{e,1}$ in step 4), one needs to compute the syndromes s_1 and s_2. It is easy to see that the complexity of this operation increases linearly in time t. This is to be expected since the code has infinite memory. A similar computational complexity also plagues the encoder, for, the encoding operation at time t is described by $c_t = G_t b_1 + \ldots + G_1 b_t$ where $\{b_i\}$ denote the source bits and hence becomes progressively hard with t.

We propose the following scheme to circumvent this problem in practice. We allow the decoder to periodically, say at $t = \ell(2T)$ ($\ell = 1, 2\ldots$) for appropriately chosen T, provide feedback to the encoder on the position of the earliest uncorrected erasure which is, say at time $t-d$. The encoder can use this information to stop encoding the source bits received prior to $t-d$, i.e., $\{b_i\}$ for $i \leq t-d-1$ starting from time $t+T$. In other words, for $\tau > t+T$, $c_\tau = G_{\tau-t+d+2} b_{t-d-1} + \ldots + G_1 b_\tau$. The decoder accordingly uses the new generator matrix starting from $t+T$. In practice, this translates to an arrangement where the decoder sends feedback at time t and can be sure that the encoder receives it by time $t+T$. Such feedback, in the form of acknowledgements from the receiver to the transmitter, is common to most packet-based modern communication and networked systems for reasonable values of T. Note that this form of feedback finds a middle ground between one extreme of having no feedback at all and another extreme where every channel output is fed back to the transmitter, the latter being impractical in most cases. The decoder proposed in Alg. 3 is easy to implement and its performance is simulated in Section 5.5.

5.2.2 Extension to Packet Erasures

The encoding and decoding algorithms presented so far have been developed for the case of bit erasures. But it is not difficult to see that the techniques generalize to the

case of packet erasures. For example, for a packet length L, what was one bit earlier will now be a block of L bits. Each binary entry in the encoding/parity check matrix will now be an $L \times L$ binary matrix. The rate will remain the same. So, at each time, k packets each of length L will be encoded to n packets each of the same length L.

Recall that the *anytime performance* of the code is determined by the delay-dependent codebook $\mathcal{C}_{t,d}$ and its distance distribution $\{N_{w,d}^t\}_{w=1}^{nd}$. In the case of packet erasures, one can obtain analogous results by defining the Hamming distance of a codeword slightly differently. By viewing a codeword as a collection of packets, define its Hamming distance to be the number of non-zero packets. The definition of the delay-dependent distance distribution $\{N_{w,d}^t\}$ will change accordingly. With this modification, one can easily apply the results developed in Sections 4.6, 4.7 and the decoding algorithm in Section 5.2 above to the case of packet erasures. For example, a reasonably simple calculation will show that a rate exponent pair (R, β) that is achievable in the case of binary erasures with bit erasure probability ϵ will be achievable in the case of packet erasures with packet length L and packet erasure probability ϵ^L. The converse is not true though and we will not delve into the calculations here.

Here we envision the anytime code operating on top of the existing packet communication layer. One can alternately consider an alternate mode where the input to the encoder is not packetized. That is, at each time, the encoder receives K bits, say, where K is not necessarily a multiple of L and uses a linear tree code to map these K bits to N bits where N is a multiple of L and corresponds to N/L packets. The rate of this code is K/N and each block in the block lower triangular generator matrix corresponding to the tree code will have dimension $N \times K$. The analysis will be no different in this case.

5.3 Decoding Over the Binary Symmetric Channel

We will first discuss a natural algorithm to sequentially perform maximum-likelihood decoding for a tree code [92]. We will then speculate on how this may be extended to the case of the binary symmetric channel.

5.3.1 A Sequential Decoder

Consider decoding an $m-$ary tree code over alphabet $\mathcal{S}$ and distance parameter α, over a discrete memoryless channel with input and output alphabet $\mathcal{S}$. Suppose that the channel introduces an error with probability ϵ, i.e., the probability that the channel reproduces the input at the output is $1-\epsilon$. Further suppose that $\epsilon < \alpha/2$. Let $r = (r_1, \ldots, r_t)$ denote the received word till time t. Let $\hat{c}_\tau$ denote the decoder's estimate of the input to the channel at time τ using channel outputs till time $t-1$. Also let $\hat{c}^{ML}_{\tau|t}$ denote the corresponding ML estimate using channel ouputs received till time t. Under the channel model assumed, maximum-likelihood estimation amounts to minimum-distance decoding, i.e.,

$$\hat{c}^{ML}_{1:t|t} = \underset{c \in \mathcal{C}}{\operatorname{argmin}} \|r - c\|$$

. One can supply a simple certificate to verify if $\hat{c}_\tau = \hat{c}^{ML}_{\tau|t}$.

Proposition 5.1. *If* $\|\hat{c}_{1:t} - r\| < \alpha t/2$, *then* $\hat{c}_{1|t} = \hat{c}^{ML}_{1|t}$

Proof. Note that

$$\|\hat{c}^{ML}_{1:t|t} - r\| \leq \|\hat{c}_{1:t} - r\| < \frac{\alpha t}{2}$$

Suppose on the contrary that $\hat{c}_1 \neq \hat{c}^{ML}_{1|t}$. Then by the tree code property

$$\|\hat{c}_{1:t} - \hat{c}^{ML}_{1:t|t}\| \geq \alpha t$$

So we have

$$\|\hat{c}_{1:t|t} - r\| = \|\hat{c}_{1:t} - \hat{c}^{ML}_{1:t|t} + \hat{c}^{ML}_{1:t|t} - r| \geq \|\hat{c}_{1:t} - \hat{c}^{ML}_{1:t|t}\| - \|\hat{c}^{ML}_{1:t|t} - r\| \geq \frac{\alpha t}{2}$$

which is a contradiction. Hence $\hat{c}_1 = \hat{c}^{ML}_{1|t}$. □

Similarly if $\|\hat{c}_{1:t|t} - r\| < \alpha t/2$ and $\|\hat{c}_{1:t|t} - r_{2:t}\| < \alpha(t-1)/2$, then $\hat{c}_1 = \hat{c}^{ML}_{1|t}$ and $\hat{c}_2 = \hat{c}^{ML}_{2|t}$. One can proceed like this until the first instant τ when $\|\hat{c}_{\tau+1:t} - r_{\tau+1:t}\| \geq$

$\alpha(t-\tau)/2$. We will state this as a lemma for easy reference.

Lemma 5.2. *Let*

$$\tau = \underset{i}{\operatorname{argmax}} \left\{ \|\hat{c}_{i:t} - r_{i:t}\| < \frac{\alpha(t-i+1)}{2} \right\}$$

Then $\hat{c}_i = \hat{c}^{ML}_{i|t}$ *for all* $1 \leq i \leq \tau$

With this observation, we are ready to describe the sequential decoder. Suppose the decoder has computed the ML estimate $\hat{c}^{ML}_{1:t-1|t-1}$ using channel outputs till time $t-1$. Extend $\hat{c}^{ML}_{1:t-1|t-1}$ by one symbol arbitrarily to get a valid guess $\hat{c}_{1:t}$, i.e., $\hat{c}_{1:t} = \left[\hat{c}^{ML}_{1:t-1|t-1}, \hat{c}_{t|t-1}\right]$ is a codeword. Use Lemma 5.2 to determine the longest prefix of $\hat{c}_{1:t}$ that can be verified to match the ML codeword and let the length of this prefix be τ, i.e., $\hat{c}^{ML}_{1:\tau|t} = \hat{c}_{1:\tau}$. The remaining portion, $\hat{c}^{ML}_{\tau+1:t}$, can be determined by an exhaustive search in the subtree of depth $t-\tau$ that is rooted at the node in the code tree that is indexed by the prefix $\hat{c}^{ML}_{1:\tau|t}$.

5.3.2 Complexity

The total number of the operations performed at time t is equal to the sum of the numbers to perform the following two tasks

1. Determining the longest prefix as in Lemma 5.2, and

2. Exhaustive search in the sub-tree

[92] shows how to perform 1) with a constant number of operations per time step. The complexity of 2) is $\mathrm{O}\big(m^{t-\tau}\big)$ since the code tree is $m-$ary. Now

$$\|\hat{c}_{\tau+1:t} - r_{\tau+1:t}\| \geq \frac{\alpha(t-\tau)}{2} \implies \|\hat{c}^{ML}_{\tau+1:t-1|t-1} - r_{\tau+1:t-1}\| \geq \frac{\alpha}{2}(t-\tau) - 1$$

The probability of this event is at most $2^{-(t-\tau-1)D(\alpha/2,\epsilon)}$. The average complexity is bounded above by

$$\sum_{\ell \geq 1} 2^{-\ell D\left(\frac{\alpha}{2},\epsilon\right)} m^{\ell}$$

which is finite provided $D(\alpha/2, \epsilon) > \log m$. One can guarantee this for small-enough rates. The more general sequential decoding algorithms in [28,47,48] guarantee finite average complexity for rates upto to the computational cutoff rate. One can rephrase the complexity distribution as follows: the probability of having to perform L operations decays as $L^{-\gamma}$ for some $\gamma > 0$. For small-enough rates $\gamma > 1$ which is when the average complexity is bounded. The same technique would apply to the linear tree codes also but linearity allows one to improve the complexity distribution to have a tail that decays as $2^{-\Omega\left(\sqrt[3]{L}\right)}$ over erasure channels which is better than any polynomial decay and performs very well in practice. Moreover the average complexity is bounded for all rates up to the channel capacity. With this background, we will speculate an approach to construct codes with similar complexity distribution over the binary symmetric channel.

5.4 Can Linear Programming Decoding Be Anytime Reliable?

We will briefly recap the fundamentals of the linear programming decoder proposed in [30] before suggesting a possible sequential approach for the causal case. Consider an arbitrary binary block code of length n and rate $R = k/n$ described by a parity check matrix $H \in \mathbb{GF}_2^{n-k \times n}$. Let $\mathcal{C} = \{c \in \mathbb{GF}_2^n \mid Hc = 0\}$ and let $r \in \mathbb{GF}_2^n$ be the vector received upon transmitting the zero codeword over a binary symmetric channel

with bit flip probability ϵ. Then the ML codeword is given by

$$\begin{aligned}
\hat{c}^{ML} &= \underset{c\in\mathcal{C}}{\operatorname{argmax}}\, p(r|c) = \underset{c\in\mathcal{C}}{\operatorname{argmax}} \sum_i \log p(r_i|c_i) \\
&= \underset{c\in\mathcal{C}}{\operatorname{argmax}} \sum_i \log p(r_i|c_i) - \log p(r_i|0) \\
&= \underset{c\in\mathcal{C}}{\operatorname{argmin}} \sum_i \left[\log \frac{p(r_i|0)}{p(r_i|1)}\right] c_i \\
&= \underset{c\in\mathcal{C}}{\operatorname{argmin}} \sum_i \gamma_i c_i, \quad \text{where} \\
\gamma_i &= \begin{cases} 1, & r_i = 0 \\ -1, & r_i = 1 \end{cases}
\end{aligned}$$

One can equivalently optimize the linear objective $\sum \gamma_i c_i$ over the convex hull of $\mathcal{C}$ which we denote with $conv(\mathcal{C})$. $conv(\mathcal{C})$ is also referred to as the codeword polytope and is contained inside the $n-$dimensional hypercube $[0,1]^n$, and includes exactly those vertices of the hypercube which are codewords. This relaxation gives the following linear program

$$\hat{c}^{ML} = \underset{f\in conv(\mathcal{C})}{\operatorname{argmin}} \sum_{i=1}^{n} \gamma_i f_i \tag{5.3}$$

So the ML codeword can be computed in principle using the linear program in (5.3). Even though one can express ML decoding as a linear program, one cannot solve it efficiently because one needs exponentially many linear inequalities to describe the polytope $conv(\mathcal{C})$. Moreover it will be miraculous if the linear program (5.3) could be solved efficiently since it is well known that ML decoding is NP-hard in general [12]. Feldman and his colleagues propose a natural relaxation to (5.3) in [30] and in [29] prove that there exist codes which under the relaxed LP can correct a constant fraction of errors. We will now describe this relaxation. The code $\mathcal{C}$ is described by the $n-k$ parity check equations each corresponding to a row of the parity check matrix H. Let $\mathcal{C}_i$ be the set of n-bit words that satisfy the i^{th} parity check equation. Then

$\mathcal{C} = \cap_{i=1}^{n} \mathcal{C}_i$ and the following is obvious

$$conv(\mathcal{C}) = conv\left(\cap_{i=1}^{n} \mathcal{C}_i\right) \subseteq \bigcap_{i=1}^{n} conv(\mathcal{C}_i) \triangleq \mathcal{P}_H \tag{5.4}$$

$\mathcal{P}_H$ is called the fundamental polytope and it contains the codeword polytope inside it. Every codeword is a vertex of $\mathcal{P}_H$ but $\mathcal{P}_H$ has additional vertices which are commonly referred to as pseudocodewords. Equation (5.4) suggests the following relaxation to (5.3)

$$\hat{f}^{LP} = \underset{f \in \mathcal{P}_H}{\operatorname{argmin}} \sum \gamma_i f_i \tag{5.5}$$

The motivation for the relaxation (5.5) is that when H is a low density parity check (LDPC, e.g., [32]) matrix, the number of inequalities required to describe $\mathcal{P}_H$ is linear in the block length n [30]. As a result, the LP in (5.5) can be solved in $\mathrm{O}\left(n^3\right)$ time.

We will need the following standard definition (e.g., [106]) before proceeding further

Definition 5.1. *The BSC pseudoweight, $\|.\|_{bsc}$, of a nonnegative $n-$dimensional vector f is defined as follows. Sort f is decreasing order of magnitude as $f^{(1)} \geq \ldots \geq f^{(n)}$. Then*

$$\|f\|_{bsc} = 2 \max \left\{ j \;\middle|\; \sum_{i \leq j} f^{(i)} < \sum_{i > j} f^{(i)} \right\}$$

Let $\mathcal{K}_H$ denote the conic hull of $\mathcal{P}_H$. Then the minimum pseudoweight, $\omega_{\min}$, is defined as

$$\omega_{\min} = \min_{f \in \mathcal{K}_H} \|f\|_{bsc} \tag{5.6}$$

To see why this definition is meaningful, assume that the zero codeword was transmitted (this is without loss of generality as proved in [30]). Then (5.5) is equivalent

to

$$\hat{f}^{LP} = \underset{f \in \mathcal{K}_H}{\operatorname{argmin}} \sum \gamma_i f_i \tag{5.7}$$

Recall that $\gamma_i = -1$ if the channel flips the i^{th} bit and $\gamma_i = 1$ otherwise. Suppose the channel flipped fewer than $\omega_{\min}/2$ bits, then $\sum \gamma_i f_i = \sum_{i:\gamma_i=1} f_i - \sum_{i:\gamma_i=-1} f_i$ which is strictly positive since $|\{i : \gamma_i = -1\}| \leq \omega_{\min}/2$. As a result, $\hat{f}_i = 0$ for all $1 \leq i \leq n$ is the unique minimum of (5.7) and hence the codeword is recovered correctly. So, $\omega_{\min}$ characterizes the number of worst case bit flips that the code can correct.

5.4.1 Sufficient Conditions

The minimum pseudoweight to LP decoding is what minimum Hamming distance is to ML decoding. If $\omega_{\min} = \alpha n$, then the LP decoder will give an error exponent whenever $\epsilon < \alpha/2$ much the same way as the ML decoder would if the minimum Hamming distance is linearly proportional to the block length. This correspondence between Hamming distance for ML decoding and $\omega_{\min}$ for LP decoding and the fact that there exist linear block codes with $\omega_{\min} = \alpha n$ (e.g., [29]) leads us to wonder if it is possible to construct causal linear codes with pseudoweight that increases linearly with delay. Recall that the $n(1-R)t \times nt$ block triangular parity check matrix $\mathbb{H}^t_{n,R}$ describes the causal code till time t (e.g., Table 4.1). Let $\mathcal{P}^t_H$ denote the fundamental polytope of $\mathbb{H}^t_{n,R}$ and $\mathcal{K}^t_H$ the conic hull of $\mathcal{P}^t_H$. Also define

$$\omega^t_{\min,d} = \min \left\{ \|f\|_{bsc} \,\middle|\, f \in \mathcal{K}^t_H, f_{t-d+1} \neq 0 \right\}$$

This definition is analogous to the definition of $w^t_{\min,d}$ in Chapter 4 where we discussed sufficient conditions on anytime distance for a causal linear code to be anytime reliable under ML decoding.

Then the following property will ensure that the code is anytime reliable under LP decoding.

Property 1. *The delay-dependent minimum pseudoweight is linearly propoertional*

to delay, i.e.,

$$\omega^t_{\min,d} = \alpha n d, \ \ \forall \ t, d \geq d_0 \tag{5.8}$$

where d_0 is a constant independent of t and d.

Note that if one performs LP decoding at every time step, then it easy to see that Property 1 guarantees anytime reliability whenever the channel bit flip probability ϵ is smaller than $\alpha/2$. So Property 1 is a sufficient condition for anytime reliability under LP. As mentioned earlier, there exists linear block codes with $\omega_{\min}$ linearly proportional to the block length [29]. Extending this to the causal case and more importantly determining if it is even possible seems to be a challenging problem. The complexity of performing LP at decoding instant t is $\mathrm{O}\big(t^3\big)$ and hence is not sustainable if the time horizon is large. In the context of distributed control, the time horizon of operation is, in principle, infinite. So for LP decoding to be plausible, we need to be able to perform it in such a way as to at least guarantee a constant average complexity at any decoding instant. Carrying the correspondence between ML decoding and LP decoding even further, it is natural to wonder if LP can be performed sequentially much the same way as ML could be as shown in Section 5.3.1. The key to performing ML sequentially is Lemma 5.2. We will need a similar certificate for the LP optimality of a pseudocodeword. Note that the linear program in (5.5) can be equivalently written as

$$\hat{f}^{LP}_{t|t} = \underset{f \in \mathcal{P}^t_H}{\operatorname{argmin}} \|r - f\|_1 \tag{5.9}$$

Central to Lemma 5.2 is a lower bound on the Hamming distance between any two codewords (i.e., the tree code property). Similarly, the sequential algorithm in Section 5.3.1 will extend to the LP case trivially if there is a similar lower bound on the ℓ_1-distance between any two pseudocodewords. We state it more precisely as follows

Property 2. *Let f, f' be any two distinct pseudocodewords (i.e., distinct vertices of*

the fundamental polytope $\mathcal{P}_H^t$). Then the following is true

$$\|f - f'\|_1 \geq \beta n(t - \tau)$$

where $\tau + 1$ is the earliest instant where they disagree, i.e., $\tau + 1 = \text{argmin}_i\{f_i \neq f_i'\}$.

If a causal linear code satisfies both Property 1 and Property 2, then it is anytime reliable under LP decoding. Furthermore, one can perform the LP sequentially in which case the probability of performing L computations at any time will decay[1] as $2^{-\Omega(\sqrt[3]{L})}$. It is a challenging open problem to examine if codes satisfies the two properties exist.

In the next Section, we present some simulations to demonstrate the efficacy of Toeplitz codes over the erasure channel.

5.5 Simulations

We present two examples and stabilize them over a binary erasure channel with erasure probability $\epsilon = 0.3$. The number of channel uses per measurement is fixed to $n = 15$. In both cases, time invariant codes $\mathbb{H}_{15,R} \in \mathbb{TZ}_{\frac{1}{2}}$, for an appropriate rate R, were randomly generated and decoded using Algorithm 3. The controller uses the Hypercuboidal filter to estimate the state.

5.5.1 Cart-Stick Balancer

The system parameters for a cart-stick balancer (also commonly called the *inverted pendulum on a cart*) with state variables of stick angle, stick angular velocity, and

[1]Note that this is modulo the partial feedback of the form assumed in Section 5.2.1 that one may need

cart velocity, when sampled with sampling duration 0.1s are (Exercise 10.15 in [31])

$$F = \begin{bmatrix} 1.161 & 0.105 & 0 \\ 3.3 & 1.161 & 0.002 \\ -3.265 & -0.160 & 0.979 \end{bmatrix}, \; G = [-0.003 \;\; -0.068 \;\; 0.859]^T, \; H = [10 \;\; 0 \;\; 0]$$

The characteristic polynomial of F is $x^3 - 3.3x^2 + 3.27x - 0.98$ and its eigenvalues are 1.75, 0.98 and 0.57. So, F is open-loop unstable. Each component of the process noise and measurement noise is i.i.d zero mean Gaussian with variance 0.01 truncated to lie in [-0.025,0.025]. The control input is given by $u_t = -K\hat{x}_{t|t}$, where $K = [-81.55 \;\; -14.37 \;\; -0.04]$. One can verify that $F - GK$ is stable. In order to apply Theorem 3.1, we write F in the following canonical form

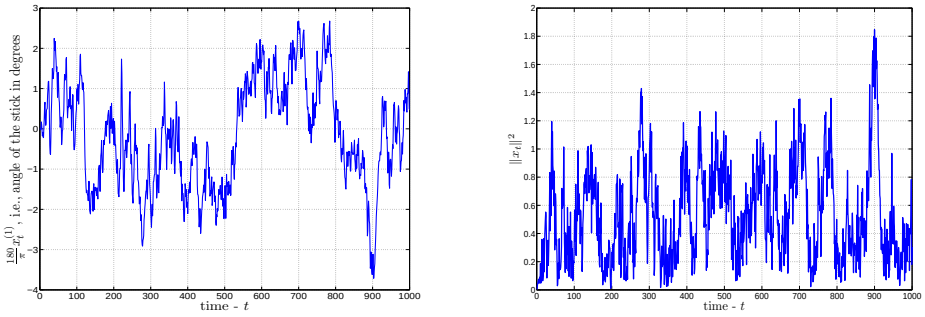
$$F_o = \begin{bmatrix} 3.3 & 1 & 0 \\ -3.27 & 0 & 1 \\ 0.98 & 0 & 0 \end{bmatrix}$$

Applying Theorem 3.1, one can stabilize x_t in the mean-squared sense provided the exponent $n\beta > 2\log\left(\rho\left(\overline{F_o}\right)\right) = 4.1035$ and the rate $nR = k > \log(3.3 + 3.27 + 0.98) = 2.1$. For $k = 5$, there exist anytime reliable codes with exponent upto $n\beta = 4.27$. Figure 5.1 plots a sample path of the above system for a randomly chosen Toeplitz code. It is clear from Figure 5.1(b) that the plant is stabilized.

5.5.2 Rate Vs. Exponent Trade-Off

This example is aimed at exploring the trade-Off between the resolution of the quantizer and the error performance of the causal code. Consider a 3-dimensional unstable system (3.4) with

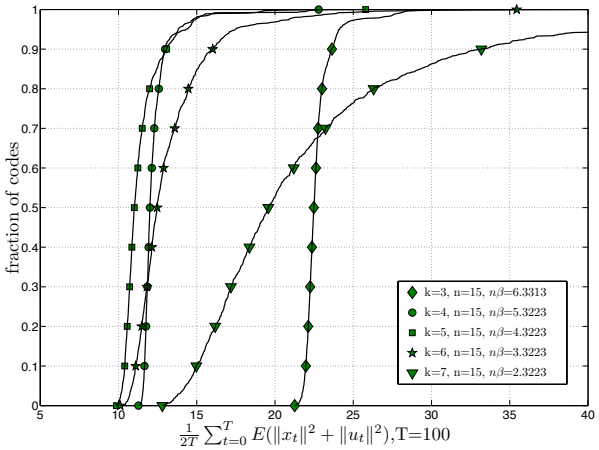
$$F = \begin{bmatrix} 2 & 1 & 0 \\ 0.25 & 0 & 1 \\ -0.5 & 0 & 0 \end{bmatrix}$$



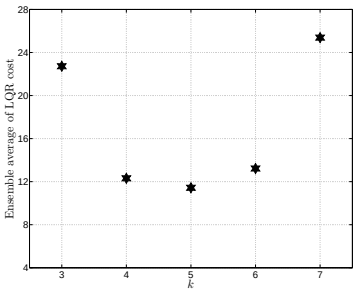
(a) The stick does not deviate by more than 3 degrees from the vertical (b) This shows that the plant is stabilized

Figure 5.1: A sample path

$G = \mathcal{I}_3$ and $H = [100]$. Each component of w_t and v_t is generated i.i.d $N(0,1)$ and truncated to [-2.5,2.5]. The eigenvalues of F are $\{2, -0.5, 0.5\}$ while $\lambda(\overline{F}) = 2.215$. The observer has access to the control inputs and we use the hypercuboidal filter outlined in Section 3.9.1. Using Theorem 3.1, the minimum required bits and exponent are given by $k = nR \geq 2$ and $n\beta \geq 2\log_2 2.215 = 2.29$. The control input is $u_t = -\hat{x}_{t|t-1}$. For $k \leq 7$, $n\beta \geq 2.32$. If $k = 8$, $n\beta = 1.32 < 2.29$. For each value of k ranging from 3 to 7, 1000 codes were generated from the ensemble $\mathbb{TZ}_{\frac{1}{2}}$. For each code, the system was simulated over a horizon of 100 time instants and the LQR cost has been averaged over 100 such runs. For a time horizon T, the LQR cost is defined as $\frac{1}{2T}\sum_{t=0}^{T}\mathbb{E}\left(\|x_t\|^2 + \|u_t\|^2\right)$. In Figure 5.2(a), the cumulative distribution function of the LQR cost is plotted for $3 \leq k \leq 7$. The $x-$axis denotes the proportion of codes for which the LQR cost is below a prescribed value, e.g., with $k = 6, n = 15$, the cost was less than 15 for 85% of the codes while with $k = 5, n = 15$, this fraction increases to more than 95%. The competition between the rate and the exponent in determining the LQR cost is evident when we look at Figure 5.2(b). When $k = 3$, the error exponent $n\beta = 6.3$ is large. So, at any time t, the decoder decodes all the source bits $\{b_\tau\}_{\tau \leq t-1}$ with a high probability. Hence, the limiting factor on the LQR cost is the resolution that the source bits b_t provide on the measurements. But when $k = 7$, the measurements are quantized to a high resolution but the decoder makes errors in decoding the source bits. So, the best choice appears to be $k = 5$.



(a) The CDF of the LQR costs for different values of the rate



(b) The LQR cost averaged over the 1000 randomly generated codes is plotted against k

Figure 5.2: The best choice of the rate is $R = 5/15 = 0.33$

Chapter 6

Simulating Protocols Over Erasure Networks

6.1 Background

We have seen in previous chapters how tree codes can be used in the context of distributed control to stabilize unstable processes over noisy channels. We have also discussed how tree codes were used to obtain an interactive analogue of Shannon's noisy channel coding theorem in [92]. The results in [92] were extended to the case of arbitrary graphs in [79] where the authors showed that tree codes can be used to simulate protocols over a group of agents connected to each other through an arbitrary directed communication graph with noisy links. They showed that one can simulate protocols with T rounds in time $\mathrm{O}\big(T\log(\Delta+1)+T\log N\big)$ with a probability of error that vanishes exponentially fast in T, where N is the number of nodes and Δ is the maximum degree. The results were presented for the case where the noisy channels were binary symmetric channels. The focus was on achieving exponentially small error probability while suffering a constant slowdown and attention was not paid to the size of the constants.

As recognized by the authors in [79], a major challenge in applying the techniques in practice was the lack of efficiently encodable and decodable constructions of tree codes. While this remains an open problem for general communication channels, we have efficient constructions for the erasure channel as discussed in Chapters 4 and

5. The erasure case allows a considerably simpler algorithm for simulating protocols. Together with the thresholds on achievable rate and exponent for linear Toeplitz codes from Chapter 4, we can obtain a tighter characterization of simulating protocols over erasure channels. We will apply these results to the problem of computing averages over graphs in Chapter 7.

6.2 Problem Setup

Consider a group of N nodes denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$. We assume that the nodes are connected by an undirected communication graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ which is often referred to as the interaction graph. Throughout the analysis $\mathcal{G}$ is assumed to be fixed and not vary with time. Let $A = [a_{ij}]$ denote the adjacency matrix of G, i.e., $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and 0 otherwise. We assume that the communication between nodes is packetized. A generic protocol over such a network of agents can be described as follows. A round of the protocol is one where every pair of neighbors exchanges one packet. Let x_t^{ij} denote the packet sent by node j to node i in round t and let $\mathcal{N}_i$ denote the neighbors of node i. Then in round $t + 1$, the packet sent by node i to a neighbor j', $x_{t+1}^{j'i}$, is a function (either deterministic or random) of the packets $\{x_{\tau \leq t}^{ij}\}_{j \in \mathcal{N}_i}$ received by node i from its neighbors up to round t. Each such round is referred to as an iteration of the protocol. Even though we treat undirected graphs here, the results trivially extend to digraphs.

We model the communication links between nodes as packet erasure links. We denote the event of successful packet reception from node j to node i at time k with the Bernoulli random variable X_k^{ij}, i.e., $X_k^{ij} = 1$ if the packet is received successfully at time k and 0 otherwise. This notation is summarized in Table 6.1. We consider two erasure models

1. *Symmetric:* $X_k^{ij} = X_k^{ij}$, and X_k^{ij}, $X_k^{m\ell}$ are independent of each other whenever $(i, j) \notin \{(m, \ell), (\ell, m)\}$, e.g., line of sight links.

2. *Asymmetric:* X_k^{ij}, $X_k^{m\ell}$ are independent of each other whenever $(i, j) \neq (m, \ell)$,

Table 6.1: Notation for Chapter 6

$\|y\|$, $y \in \mathbb{R}^N$	$\sqrt{\sum_{i=1}^{N} y_i^2}$, i.e., the two norm of y
$\mathcal{N} = \{1, 2, \ldots, N\}$	the set of nodes
$\mathcal{G} = (\mathcal{N}, \mathcal{E})$	the underlying communication graph
$A = [a_{ij}]$	the adjacency matrix of G, i.e., $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and 0 otherwise
$\mathcal{N}_i$	the set of neighbors of node i, i.e., $\mathcal{N}_i = \{j' \| a_{ij'} = 1\}$
Δ	largest degree of any vertex in G
p	packet erasure probability
X_k^{ij}	1 if the packet sent from node j to node i at time k is successfully received and 0 o.w
$D(p, q)$	$p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$ i.e., Kullbeck Leibler divergence between Bernoulli(p) and Bernoulli(q)

in particular X_k^{ij} and X_k^{ji} are independent, e.g., wireless links.

6.3 Symmetric Link Failures

Note that the underlying interaction graph $\mathcal{G}$ is fixed while each link is modeled as a packet erasure channel. The graph $\mathcal{G}$ is assumed to be connected and the links are undirected. If all agents know that link failures are symmetric, then each link is effectively a packet erasure channel with feedback. In each communication round, node i would know that its packet transmission to node j is erased if it receives an erasure from node j in the same round. We now define the communication protocol.

6.3.1 Protocol Implementation

A communication round is defined as one in which every node in the graph transmits one packet to each of its neighbors. The nodes are said to have completed m iterations if all of them successfully computed m iterations of the protocol. Note that this will

in general take more than m communication rounds. Since each link is effectively an erasure channel with feedback, the optimal communication scheme at each node is to retransmit until successful reception. We describe this more precisely as follows. Let e denote an erasure. For each edge $j \to i$, we associate an input queue, Q_{in}^{ij}, and an output queue, Q_{out}^{ij}. $Q_{in,t}^{ij}$ contains the packets transmitted by node j to node i up to and including communication round t while $Q_{out,t}^{ij}$ contains the packets received by node i from node j (e.g., Figure 6.3.1).

Also let b_t^{ij} denote the packet transmitted by node j to node i in communication round t and let z_t^{ij} denote the received packet. Then

$$z_t^{ij} = \begin{cases} b_t^{ij} & \text{w.p } 1-p \\ e & \text{w.p } p \end{cases} \tag{6.1}$$

Now if $z_t^{ji} = e$, then node j infers that b_t^{ij} was erased and hence retransmits it in the next communication round unless b_t^{ij} was a 'wait' symbol which we describe as follows. We say that a node i has 'new data' if it could compute one or more new iterations of the protocol. During communication rounds where node j does not have any new data to transmit, it transmits a wait symbol which we denote with w. The transmission from node i to node j in round t is described in Algorithm 4.

Algorithm 4 Node i's transmission to node j in round t

if $z_{t-1}^{ji} = e$ **and** $b_{t-1}^{ji} \neq w$ **then**
  $b_t^{ji} = b_{t-1}^{ji}$, i.e., retransmit
else
  For each $j' \in \mathcal{N}_i$, let $\ell_{t,j'} = \max\{\ell' \mid x_{\ell'}^{ij'} \in Q_{out,t}^{ij'}\}$
  Compute $\ell_t = \min_{j' \in \mathcal{N}_i} \ell_{t,j'}$
  if $\ell_t = \ell_{t-1} + 1$ **then**
    Compute $x_{\ell_t+1}^{ji}$ using the protocol and set $b_t^{ji} = x_{\ell_t+1}^{ji}$ (note that $\ell_t \leq \ell_{t-1}+1$) for all $j \in \mathcal{N}_i$
  else
    (i.e., $\ell_t = \ell_{t-1}$) set $b_t^{ji} = w$
  end if
end if

The algorithm is illustrated through an example in Figure 6.1. Using such an

algorithm, we have the following result.

Theorem 6.1. *Let $P_{M,R'}$ denote the probability that the network requires more than M communication rounds to compute MR' iterations of the protocol. Further suppose that the packet erasure probability is p and that erasures are symmetric. Then*

$$P_{M,R'} \leq N2^{-M(D(1-R',p)-\log(\Delta+1))} \tag{6.2}$$

In particular, whenever R' satisfies

$$D(1-R',p) > \log(\Delta+1) \tag{6.3}$$

$P_{M,R'}$ decays exponentially fast in M. Recall that Δ the maximum degree.

Proof. See Appendix 6.6.1. □

The statement of Theorem 6.1 suggests a natural definition of the rate of the simulation (i.e., Algorithm 4), $R_s(p)$, as follows

$$R_s(p) = \sup\left\{R' > 0 \mid \lim_{M\to\infty} P_{M,R'} = 0\right\} \tag{6.4}$$

Then Theorem 6.1 can be rephrased as $R_s(p) \geq R(p)$, where

$$R(p) \triangleq \sup_{R'\geq 0}\{R' \mid D(1-R',p) > \log(\Delta+1)\} \tag{6.5}$$

The proof technique is inspired by the technique used in [79]. We use the simpler erasure model of communication to improve upon the thresholds one can obtain by directly applying the technique in [79].

Note that that $R(p) > 0$ if and only if $p < 1/(1+\Delta)$. This means that the proof technique used here does not allow us to prove successful protocol simulation if the erasure probability is larger than $1/(1+\Delta)$. We can demonstrate how to overcome this. In fact, one can show that simulation will be successful with high probability for all $0 \leq p \leq 1$, we will state the result as follows.

Theorem 6.2. *Let $P_{M,R'}$ denote the probability that the network requires more than M communication rounds to compute MR' iterations of the protocol. Further suppose that the packet erasure probability is p and that erasures are symmetric. Then*

$$P_{M,R'} \leq N2^{-MD(R',(1-p)^{|\mathcal{E}|})} \tag{6.6}$$

In particular, whenever R' satisfies

$$R' < (1-p)^{|\mathcal{E}|} \tag{6.7}$$

$P_{M,R'}$ decays exponentially fast in M. Recall that N is the number of nodes and $|\mathcal{E}|$ is the number of edges in the network.

Proof. See Appendix 6.6.3 □

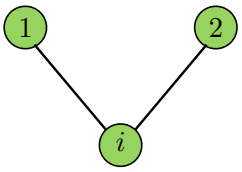
Theorem 6.2 is equivalent to $R_s(p) \geq (1-p)^{|\mathcal{E}|}$. Putting the Theorems 6.1 and 6.2 together, we have

$$R_s(p) \geq \max\left\{R(p), (1-p)^{|\mathcal{E}|}\right\} \tag{6.8}$$

The lower bounds on $R_s(p)$ obtained in Theorems 6.1 and 6.2 are qualitatively very different. While the latter depends on the number of edges, the former depends only the degree of the graph. It will be interesting to unify the proof techniques for the two Theorems to get a single tight lower bound on $R_s(p)$, better even compute it.

6.4 Asymmetric Link Failures and Tree Codes

Now suppose packet erasures are not symmetric. Then a repetition code is not applicable because a node does not know if its transmissions were decoded successfully or not. Here, we will need to use tree codes. In Section 6.2, we assumed that an iteration of the protocol corresponds to every pair of nodes exchanging one packet each. In general, it could be more than one packet, say it is k. We will encode it using a tree



(a)

$Q^{i,1}_{out,5}$:	x_0^{i1}	x_1^{i1}	e	e	x_2^{i1}
$Q^{i,2}_{out,5}$:	x_0^{i2}	e	x_1^{i2}	w	x_2^{i2}
$Q^{1,i}_{in,5}$:	x_0^{1i}	x_1^{1i}	w	x_2^{1i}	x_2^{1i}
$Q^{2,i}_{in,5}$:	x_0^{2i}	x_1^{2i}	x_1^{2i}	x_2^{2i}	w
	past $\longleftarrow$			$\longrightarrow$ present	

(b)

Figure 6.1: Consider an instance of the queues at node i. Suppose its only neighbors are nodes 1 and 2. In round 2, node i receives an erasure from node 2 and infers that its own transmission to node 2 must also have been erased. As a result, node i retransmits x_1^i to node 2 in round 3. Similarly in round 3, node i knows that its transmission to node 1 was erased. Since the erased symbol was only a 'wait', node i does not retransmit it in round 4. Instead, it checks if it can perform another iteration of the protocol. In this case, it can and hence transmits the new data x_2^i to node 1. In round 5, node i does not have any new data to transmit to node 2 and hence transmits a 'wait'.

Figure 6.2: Input and Output queues on an edge

code to output n packets at each time (e.g., Section 5.2.2). Let the packet length be Λ. The rate of the code is $r = k/n$. Here, one round of communication corresponds to every pair of neighbors exchanging n packets each. Then under the asymmetric erasure model, node i does not known which of the n transmitted packets have been received by each of its neighbors in each communication round.

Consider the pair of nodes i, j and let b_t^{ji} denote the t^{th} information packet destined to node j from node i. Then the data actually transmitted by node i is given by

$$c_\ell^{ji} = \sum_{\ell'=1}^{\ell} G_{\ell'} b_{\ell'}^{ji} \tag{6.9}$$

Suppose that the code is $(r, \beta, d_o)-$anytime reliable so that we have $P(\hat{b}_{\ell'|\ell}^{ji} \neq b_{\ell'}^{ji}) \leq 2^{-n\Lambda\beta(\ell-\ell')}$ for all $\ell - \ell' \geq d_o$. We will further assume that $d_o = 0$. This does not change the results qualitatively and will comment on its effect in Section 6.4.1. Let the unnormalized exponent be $\beta' = n\Lambda\beta$.

Since the channel is an erasure channel, the maximum-likelihood decoder amounts to solving linear equations. This can be done recursively and efficiently as shown in Chapter 5. Whenever the equations admit a unique solution to some of the variables, those variables are correctly decoded. We leave the remaining variables as erasures and do not venture a guess about their value. As a result, the decoder always knows whenever it decodes something correctly.

6.4.1 Protocol Implementation

Like the case of repetition coding for symmetric erasures, for each link $j \to i$, we associate two queues $Q_{in,t}^{ij}$ and $Q_{out,t}^{ij}$ although with a slightly different meaning. The queue $Q_{in,t}^{ij}$ contains all the information packets transmitted by node j to node i till round t. In other words, $Q_{in,t}^{ij} = \{b_\tau^{ij}\}_{\tau \leq t}$. On the other hand, $Q_{out,t}^{ij}$ are node i's estimates of the information packets transmitted by node j so far, i.e., $Q_{out,t}^{ij} = \{\hat{b}_{\tau|t}^{ij}\}_{\tau \leq t}$. With this setup, the mechanics of the protocol is very simple and is outlined in Algorithm 5.

Algorithm 5 Node i's transmission to its neighbors in round t

1: For each $j' \in \mathcal{N}_i$, compute $\ell_{t,j'} = \max\{\ell' \mid x_{\ell'}^{ij'} \in Q_{out,t}^{ij'}\}$ and let $\ell_t = \min_{j' \in \mathcal{N}_i} \ell_{t,j'}$
2: Also compute $m_{t,j'} = \max\{m' \mid x_{m'}^{j'i} \in Q_{in,t}^{j'i}\}$ and let $m_t = \min_{j' \in \mathcal{N}_i} m_{t,j'}$
3: **if** $\ell_t + 1 > m_{t-1}$ **then**
4: Compute $x_{m_{t-1}+1}^{ji}$ using the protocol and set $b_t^{ji} = x_{m_{t-1}+1}^{ji}$ for all $j \in \mathcal{N}_i$
5: **else**
6: set $b_t^{ji} = w$ for all $j \in \mathcal{N}_i$
7: **end if**

We can now compute bounds on the slowdown of the above simulation algorithm and we state it as the following theorem.

Theorem 6.3. *Let $P_{M,R'}$ denote the probability that the network requires more than M communication rounds to compute MR' iterations of the protocol. Further suppose that the packet erasure probability is p and that erasures are asymmetric. Suppose each node uses a $(R, \beta)-$anytime reliable code. Then*

$$P_{M,R'} \leq N 2^{-M\left((1-R')\frac{\beta'}{2} - H(R') - \log(\Delta+1)\right)} \tag{6.10}$$

In particular, whenever R' satisfies

$$(1 - R')\beta'/2 > H(R') + \log(\Delta + 1) \tag{6.11}$$

$P_{M,R'}$ decays exponentially fast in M.

Proof. See Appendix 6.6.2 □

Recall that r is the rate of tree code used. Then analogous to (6.4), we can define the rate of the simulation described in Algorithm 5, $R_a(r, p)$, as follows

$$R_a(r, p) = \sup\left\{R' > 0 \,\middle|\, \lim_{M \to \infty} P_{M,R'} = 0\right\} \tag{6.12}$$

where $P_{M,R'}$ is as defined in Theorem 6.3. The Theorem is then equivalent to $R_a(p) \geq$

$\rho(r,p)$ where

$$\rho(r,p) \triangleq \sup_{R' \geq 0} \{R' \,|\, (1-R')\beta'/2 > H(R') + \log(\Delta+1)\} \tag{6.13}$$

Note that the notation $\rho(r,p)$ is justified because β' is a function of the code rate r. Then much like Section 6.3, it is easy to see that $\rho(r,p) > 0$ if and only if $\beta' > 2\log(1+\Delta)$. Recall that $\beta' = n\Lambda\beta$. So we can guarantee $n\Lambda\beta > 2\log(1+\Delta)$ by choosing an appropriately large n. It is interesting to note that much like in control, it appears that we need a large enough exponent β' in order to be able to simulate protocols with a constant slowdown. Although it is not clear that this is necessary.

We had earlier assumed that we are given an $(r,\beta,d_o)-$anytime reliable code for $d_o = 0$. A positive d_o affects the slowdown only by a constant factor. In other words, Theorem 6.3 goes through by re-defining $P_{M,R'}$ as the probability that the network requires more than M communication rounds to compute MR'/d_o iterations of the protocol. So, almost any code in the Toeplitz ensemble will guarantee successful simulation with exponentially small error probability while suffering a constant slowdown. The effect of d_o is then to reduce the simulation rate by a factor $1/d_o$.

6.4.2 Comparison to Literature

The only point of comparison to the this type of result is the analysis in [79] which deals with the binary symmetric channel. The proof technique there also applies to the erasure case but the resulting bounds, as we will briefly argue, will be weaker[1] than what was obtained in Theorem 6.3. For simplicity, consider the case when the communication channels between nodes are binary erasure links with erasure probability p. Also suppose that each communication round consists of exchanging n bits between every pair of neighbors and that the rate of the tree code used at every node is r. We will state here without proof that in such a setup the technique in [79]

[1]We must note though that this comparison is not completely fair since the erasure channel being a simpler model than the binary symmetric channel admits tighter analysis.

gives the following lower bound, $\rho_S(r,p)$, on $R_a(r,p)$

$$\rho_S(r,p) = \sup_{R'>0} \left\{ R' \;\middle|\; \frac{n}{2} D\left(H^{-1}(1-r), p\right) - \log(\Delta+1) > \frac{H(R') + \log(\Delta+1)}{(1-R')} \right\} \tag{6.14}$$

The corresponding lower bound for the analysis presented here is

$$\rho(r,p) = \sup_{R'>0} \left\{ R' \;\middle|\; (1-R')\frac{n}{2}\beta > H(R') + \log(\Delta+1) \right\} \tag{6.15}$$

One can use Theorem 4.8 to compute β as a function of r and observe that the resulting expression for β satisfies

$$\beta > D\left(H^{-1}(1-r), p\right)$$

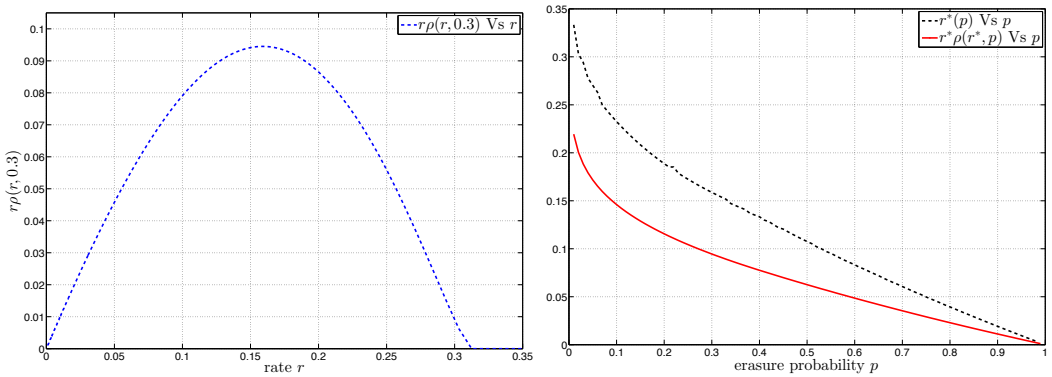
which from (6.14) and (6.15) implies that $\rho(r,p) > \rho_S(r,p)$.

6.4.3 Code Rate Vs Simulation Rate

Note that $R_a(r,p)$ is the slowdown in the number of rounds of communication and does not take into account the length of each round. Due to coding, the length of each communication round is now larger due to the larger number of packets exchanged. More precisely, the total number of packets exchanged in order to simulate T iterations of the protocol when there is coding is approximately $nT/R_a(r,p)$. On the other hand, when communication is noiseless, T iterations requires exchanging nrT packets. The overall slowdown, $R_c(r,p)$, of this simulation is then given by

$$R_c(r,p) = rR_a(r,p) \tag{6.16}$$

A lower bound on $R_c(r,p)$ is given by $R_c(r,p) \geq r\rho(r,p)$. As r increases, it is easy to see that $\rho(r,p)$ decreases. In practice, one should choose a rate r that maximises $R_c(r,p)$. Given that we only have lower bounds on $R_c(r,p)$, we can choose the rate r that maximizes $r\rho(r,p)$. This trade-off is very similar to the trade-off between the



(a) The best choice of the rate r is 0.159, i.e., $r^*(0.3) = 0.159$

(b) Plot of $r^*(p)$ and overall rate of the simulation $r^*(p)\rho\left(r^*(p), p\right)$

Figure 6.3: Trade-Off between code rate and overall simulation rate

rate and exponent observed in the context of coding for control (e.g., Section 3.7.2). We will demonstrate through a simple simulation. Consider the binary erasure case (i.e., packet length $\Lambda = 1$) with $k = 10$ and a graph with maximum degree $\Delta = 31$. Then the rate r can be adjusted by changing n. For a given erasure probability p, the rate r that maximizes $r\rho(r, p)$, denote $r^*(p)$, is numerically computed and the results are plotted in Figure 6.4.3. The trade-off between coding rate and simulation rate is clear in Figure 6.4.3 which plots $r\rho(r, 0.3)$ as a function of r.

6.5 Summary

Motivated by the availability of efficiently decodable linear tree codes, we considered the problem of simulating protocols over erasure networks. We considered two erasure models, symmetric (e.g., line of sight) and asymmetric (e.g., wireless). Symmetric and asymmetric erasure models correspond to erasure channels with and without feedback respectively. We use repetition codes in the symmetric case and tree codes in the asymmetric case to simulate protocols with an exponentially small error probability in the protocol length while suffering a constant slowdown, which we call the rate of the simulation. We obtain novel lower bounds on the rate of the simulation and argue that they improve upon those in the literature. We also comment on the trade-off

between the rate of the tree code and the rate of the simulation and note that it is very similar to the trade-off between the rate and exponent in the context of control. In the next Chapter, we apply the simulation algorithms developed and studied here to the problem of average consensus.

6.6 Appendices

6.6.1 Proof of Theorem 6.1

We will begin by identifying the state of the protocol in Algorithm 4. For the sake of clarity, we will refer to nodes using letters u, v, etc., instead of i, j. Recall that $\mathcal{N}_v$ denotes the set of neighbors of v. For each node v at time t (i.e., after round t), we associate $|\mathcal{N}_v|$ variables $\{n_{vu}(t)\}_{u\in\mathcal{N}_v}$, where $n_{vu}(t)$ denotes the latest iterate of node u that is available to node v at time t. In other words, $n_{vu}(t)$ is the largest integer τ such that x_τ^{vu} is available to node v. We further define

$$n_v(t) \triangleq 1 + \min_{u\in\mathcal{N}_v} n_{vu}(t) \tag{6.17}$$

Note that $n_v(t)$ is the latest iteration of the protocol that node v can compute at time t. In other words, node v has computed $\{x_\tau^{uv}\}_{\tau\leq n_v(t)}$ for all $u \in \mathcal{N}_v$ and no more. With this setup, it is clear that Algorithm 4 would have executed $\min_v n_v(t)$ iterations of the protocol till time t. Note that the rate of the protocol is then given by $R = \lim_{t\to\infty} \frac{min_v n_v(t)}{t}$, which is a random variable for a specific run of the protocol. We now state the evolution of $n_{vu}(t)$ as a lemma below.

Lemma 6.4. *Let $X_t^{vu} = 1$ if the edge (v, u) is erased in round t and 0 otherwise. Then the evolution of $n_{vu}(t)$ is given by the following equation*

$$n_{vu}(t+1) = n_{vu}(t) + X_{t+1}^{vu}\mathbb{1}_{[n_u(t)>n_{vu}(t)]} \tag{6.18}$$

Proof. The proof follows from the following simple observations

1. $n_{vu}(t)$ increases by at most 1 in each step
2. In any round, if node u receives an erasure on a link, it will infer that its transmission on that link was also erased. As a result, node u has knowledge of $n_{vu}(t)$ at all times t

3. In round $t+1$, if either the edge (v,u) is erased or node u sends a w to node v, then $n_{vu}(t+1) = n_{vu}(t)$

4. Node u sends a 'wait' w to node v in round $t+1$ if and only if $n_{vu}(t) = n_u(t)$.

. □

We say that round t got wasted at node v if $n_v(t-1) = n_v(t)$, i.e., node v could not perform a new iteration of the protocol at time t. The proof idea is as follows: for each node v at time t, we will argue that there exists a sequence of t edges of which at least $t - n_v(t)$ edges have failed. We then union bound over all possible choices of such t edges.

Before proceeding further, we define an object which we call the 'trellis', for lack of a better word. Associated to any undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represented by the adjacency matrix A, we define an infinite trellis $\mathcal{T}(\mathcal{G}) = (\mathcal{V}_\mathcal{T}, \mathcal{E}_\mathcal{T})$ as follows. Associated to each node v in $\mathcal{V}$, there are countably infinitely many copies $\{_k\}_{k\geq 0}$ in $\mathcal{V}_\mathcal{T}$. Let I denote a $|\mathcal{V}| \times |\mathcal{V}|$ identity matrix. Then the nodes $\mathcal{V}_\mathcal{T}$ and edges $\mathcal{E}_\mathcal{T}$ of $\mathcal{T}(\mathcal{G})$ are given by

$$\mathcal{V}_\mathcal{T} = \bigcup_{v\in\mathcal{V}} \bigcup_{k\geq 0} \{v_k\} \tag{6.19a}$$

$$\mathcal{E}_\mathcal{T} = \{(v_\tau, u_{\tau'}) \mid |\tau - \tau'| = 1, (A+I)_{vu} = 1\} \tag{6.19b}$$

The edges in $\mathcal{E}_\mathcal{T}$ are all undirected, i.e., (u_0, v_1) and (v_1, u_0) are treated as a single edge. The trellis for an example network is given in Figure 6.4.

Definition 6.1 (time-like)**.** *Any sequence of edges (or a path), $\mathcal{S}_t$, in the trellis $\mathcal{T}(\mathcal{G})$ of the type*

$$\mathcal{S}_t = \left\{(v_t, u_{t-1}^{(t-1)}), (u_{t-1}^{(t-1)}, u_{t-2}^{(t-2)}), \ldots, (u_1^{(1)}, u_0^{(0)})\right\}$$

will be called 'time-like' ending in node v_t

An edge $(u_\tau^{(\tau)}, u_{\tau-1}^{(\tau-1)}) \in \mathcal{E}_\mathcal{T}$ is said to be erased if there was an erasure on the edge $(u^{(\tau)}, u^{(\tau-1)}) \in \mathcal{E}$ in round τ. The time-like sequence $\mathcal{S}_t$ is said to have ℓ erasures if ℓ of the t edges in $\mathcal{S}_t$ were erased. We are now ready to state the key lemma from which the proof of Theorem 6.1 follows easily.

Lemma 6.5. *If after t rounds of communication, node v has performed $n_v(t)$ iterations of the protocol, then there exists a time-like sequence of t edges ending in node v_t that have at least $t - n_v(t)$ erasures among them.*

We will first prove Theorem 6.1 using Lemma 6.5. Suppose after t communication rounds, node v performed Rt iterations of the protocol, for some $R < 1 - p$. Recall that the probability of an erasure is p. Then there must be a time-like sequence of t edges with at least $(1 - R)t$ erasures, the probability of which is approximately $2^{-tD(1-R,p)}$, where $D(q, p) = q \log(q/p) + (1 - q)log(1 - q/1 - p)$. Now there are at most $(\Delta + 1)^t$ choices of such time-like sequences. Then, doing a union bound over all these sequences and over all nodes, we get

$$P_{R,t} \leq N(\Delta + 1)^t 2^{-tD(1-R,p)} \tag{6.20}$$

where $P_{R,t}$ is the probability that the network performed Rt or fewer iterations of the protocol in t rounds and N is the number of nodes in the network. This is the claim in Theorem 6.1. We will now prove the lemma.

Proof of Lemma 6.5. For ease of presentation, we will introduce the following notation in the rest of the proof.

a) *we will refer to any time-like sequence of τ edges ending in v_τ that has $\tau - n_v(\tau)$ or more erasures as a "witness" at v_τ.*

b) *We will call a node $u \in \mathcal{N}_v$ a "bottleneck" for node v in round t iff $n_{vu}(t-1) = n_v(t-1) - 1$, i.e., $n_{vu}(t-1) = min_{u' \in \mathcal{N}_v} n_{vu'}(t-1)$.*

The lemma claims that there is a witness at v_t for all $v \in \mathcal{V}$ and $t \geq 0$. We will prove this by induction. The hypothesis is clearly true for $t = 0$. Suppose it is true

for all nodes $v \in \mathcal{V}$ and all $\tau \leq t-1$. Recall that we say that round t at node v is wasted only if $n_v(t-1) = n_v(t)$. There are two broad cases, round t gets wasted at node v or it does not.

1) Suppose round t is not wasted, i.e., $n_v(t) = n_v(t-1) + 1$. Then by the induction hypothesis, there is a witness at v_{t-1}. Appending the edge (v_{t-1}, v_t) to this witness gives us a witness for v_t.

2) It remains to consider the case where round t gets wasted at node v, i.e., $n_v(t) = n_v(t-1)$.

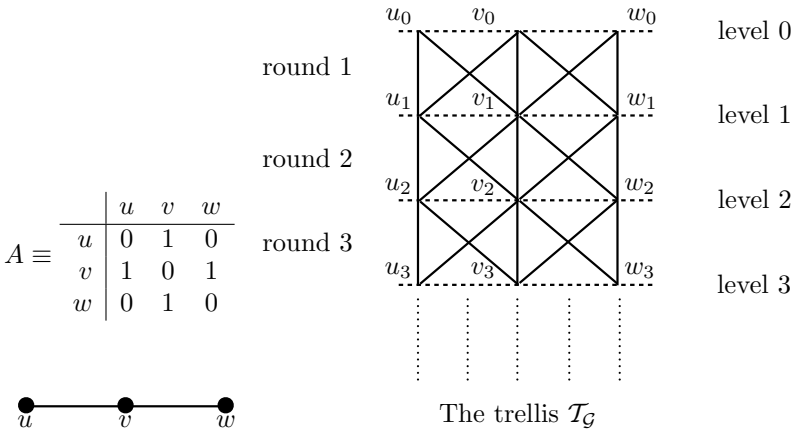
We will divide case 2) above into two subcases: a) $\exists$ a $u \in \mathcal{N}_v$ s.t $n_u(t-1) = n_v(t-1) - 1$ and b) such a neighbor does not exist.

a) If there is a neighbor $u \in \mathcal{N}_v$ such that $n_u(t-1) = n_v(t-1) - 1$, then the witness for v_t is obtained by appending the edge (v_t, u_{t-1}) to the witness at u_{t-1}.

b) Here $n_u(t-1) \geq n_v(t-1)$ for all $u \in \mathcal{N}_v$. Since $|n_u(\tau) - n_v(\tau)| \leq 1$ for any τ, we can partition the neighbors of v into two classes $Y = \{u \in \mathcal{N}_v \mid n_u(t-1) = n_v(t-1)\}$ and $Z = \{u \in \mathcal{N}_v \mid n_u(t-1) = n_v(t-1) + 1\}$. Furthermore, let $B = \{u \in \mathcal{N}_v \mid n_{vu}(t-1) = n_v(t-1) - 1\}$ denote the bottlenecks for v in round t.

We will further divide case b) above into two subcases: i) $B \cap Z = \emptyset$ and ii) $B \cap Z \neq \emptyset$

i) $B \cap Z = \emptyset$, i.e., there are no bottlenecks in the set of neighbors Z. Observe that a bottleneck neighbor will not send a wait w. Also for any $u \in B \cap Y$, $n_{vu}(t-1) = n_v(t-1) - 1 = n_u(t-1) - 1$. So, the data transmitted by node u to node v in round t is $x^{vu}_{n_u(t)}$, i.e., iteration $n_u(t)$ of the protocol. Since round t at node v got wasted, at least one of the edges to a bottleneck neighbor must have been erased in round t. Otherwise, node v would have been able to compute a new iteration of the protocol and the round would not have been wasted. Suppose the erasure happened on edge (v, u) for some $u \in B \cap Y$. Then appending edge (v_t, u_{t-1}) to the witness at u_{t-1} will give us the witness at v_t.



(a) An example network (b) Trellis associated to the network in (a)

Figure 6.4: This depicts the trellis associated to a network of three nodes connected in a straight line. The thick lines represent edges.

ii) $B \cap Z \neq \emptyset$, i.e., there is a neighbor $u \in B \cap Z$ such that $n_u(t-1) = n_v(t-1)+1$ and $n_{vu}(t-1) = n_v(t-1) - 1 = n_u(t-1) - 2$. Furthermore, there must be a neighbor $u \in B \cap Z$ whose transmission to v in round t must have been erased (else there must be an edge to $B \cap Y$ which was erased and we revert back to case i)). Note that $n_u(t-2) \geq n_v(t-1)$. It follows from Lemma 6.4 that node u must have transmitted iteration $n_v(t-1)$ in round $t-1$ as well as round t and both were erased since $n_{vu}(t) = n_{vu}(t-1) = n_v(t-1) - 1$. Since this erasure model considers symmetric erasures, the transmission from v to u in round $t-2$ is also erased. Appending the edges (v_t, u_{t-1}) and (u_{t-1}, v_{t-2}) to the witness at v_{t-2} gives us the witness for v_t.

This completes the proof of Lemma 6.5.

□

6.6.2 Proof of Theorem 6.3

We will begin the proof with three preliminary results before moving to the main argument. Recall that an $(R, \beta)-$anytime reliable code is one that guarantees

$$P\left(\hat{b}_{\tau|t} \neq b_\tau\right) \leq 2^{-n\Lambda\beta(t-\tau+1)}$$

where n is the number of packets transmitted in each communication round and Λ is the packet length. To avoid clutter, we define $\beta' = n\Lambda\beta$. For such a code that is linear, we can say the following.

Lemma 6.6. *Suppose $\{b_i\}_{i\geq 0}$ are encoded and decoded using a causal linear $(R, \beta)-$anytime reliable code. Consider the following events*

$$Y(\tau_1', \tau_1): \quad \tau_1' = 1 + \operatorname*{argmax}_{\ell}\{\hat{b}_{\ell|\tau_1} = b_\ell\}$$
$$Y(\tau_2', \tau_2): \quad \tau_2' = 1 + \operatorname*{argmax}_{\ell}\{\hat{b}_{\ell|\tau_2} = b_\ell\}$$

i.e., $Y(\tau_i', \tau_i)$ is the event that at decoding instant τ_i, the position of the earliest error is at τ_i' for $i = 1, 2$. Furthermore, suppose that the intervals $[\tau_1', \tau_1]$ and $[\tau_2', \tau_2]$ are disjoint. Then we have

$$P\left(Y(\tau_1', \tau_1) \cap Y(\tau_2', \tau_2)\right) \leq 2^{-\beta'\left(|\tau_1 - \tau_1' + 1| + |\tau_2 - \tau_2' + 1|\right)} \tag{6.21}$$

The probability above is only over the randomness of the channel.

Proof. Without loss of generality, assume that $\tau_2' > \tau_1$. Due to linearity, we can assume without losing generality that the input $b_i = 0$ for $i \geq 0$. Let E_i denote the portion of the erasure pattern introduced by the channel during the interval $[\tau_i', \tau_i]$ that resulted in the event $Y(\tau_i', \tau_i)$. Then, we claim that $P(E_i) \leq 2^{-\beta'|\tau_i - \tau_i' + 1|}$. This follows from the simple observation that if the encoder input in the first $\tau_i - \tau_i' + 1$ instants is all zero and the corresponding channel erasure pattern is E_i, then $Y(\tau_i', \tau_i)$ implies that at the decoding instant $\tau_i - \tau_i'$, the earliest error would have happened at time 0, the probability of which is at most $2^{-\beta'|\tau_i - \tau_i' + 1|}$.

Since the intervals $[\tau_1', \tau_1]$ and $[\tau_2', \tau_2]$ are disjoint, the erasure patterns E_1 and E_2 correspond to independent channel uses. So we have

$$P\left(Y(\tau_1', \tau_1) \cap Y(\tau_2', \tau_2)\right) \leq P(E_1, E_2) = P(E_1)P(E_2)$$

The result now follows. □

For ease of presentation, we introduce the following definition

Definition 6.2 (Error Interval)**.** *With respect to the notation in Lemma 6.6, we refer to the interval $[\tau_i', \tau_i]$ as the error at time τ_i.*

Before proceeding with the rest of the proof, we will recall a lemma from [92] and state it here for easy reference.

Lemma 6.7 (Lemma 7, [92])**.** *In any finite set of intervals on the real line whose union J is of total length s there is a subset of disjoint intervals whose union is of total length at least $s/2$*

We will now state a version of Lemma 6.6 when the error intervals are not necessarily disjoint.

Lemma 6.8. *If $\{b_i\}_{i\geq 0}$ are encoded and decoded using a causal linear $(R, \beta)-$anytime reliable code, then*

$$P\left(\hat{b}_{\tau_1'|\tau_1} \neq b_{\tau_1'}, \ldots, \hat{b}_{\tau_m'|\tau_m} \neq b_{\tau_m'}\right) \leq 2^{-\frac{\beta'(\sum_i |\tau_i - \tau_i' + 1|)}{2}}$$

Proof. The proof follows directly from Lemma 6.6 and Lemma 6.7. □

We use an argument very similar to the one used in proving Theorem 6.1. We will define a trellis $\vec{\mathcal{T}}(\mathcal{G})$ exactly the same way we defined $\mathcal{T}(\mathcal{G})$ except that the edges $\vec{\mathcal{E}}_{\mathcal{T}}$ are now directed and they point forward in time, i.e., downwards w.r.t to the Figure 6.4(b). In other words, for neighbors $(u, v) \in \mathcal{V}$, the edge (v_t, u_{t-1}) is directed from node u_{t-1} to node v_t and represents the transmission from u to v in round t.

Recall the definition of a time-like sequence of edges, $\mathcal{S}_t$, from Definition 6.1. Let

$$\mathcal{S}_t = \left\{(u_t^{(t)}, u_{t-1}^{(t-1)}), (u_{t-1}^{(t-1)}, u_{t-2}^{(t-2)}), \ldots, (u_1^{(1)}, u_0^{(0)})\right\}$$

Let B_τ be the error interval at decoding instant τ on the edge node $(u^{(\tau)}, u^{(\tau-1)}) \in \mathcal{E}$. We alternately call B_τ the error interval on the edge $(u_\tau^{(\tau)}, u_{\tau-1}^{(\tau-1)}) \in \vec{\mathcal{E}}_{\mathcal{T}}$. Then we define $|\mathcal{S}_t|$ as follows

$$|\mathcal{S}_t| = \sum_{(v,u)\in\mathcal{E}} |B_{vu}|, \text{ where} \tag{6.22}$$

$$B_{vu} = \bigcup_{\tau:(u^{(\tau)},u^{(\tau-1)})=(v,u)} B_\tau \tag{6.23}$$

This definition is motivated by the fact that the packet erasure events during an error interval on a given edge, say $(v, u) \in \mathcal{E}$, are independent of those in an error interval on a different edge $(v', u') \neq (v, u)$ in any round of communication. So, intuitively $|\mathcal{S}_t|$ captures the number of independent "bad" channel realizations seen by the edges in $\mathcal{S}_t$. In what follows, we will show a connection between the number of wasted communication rounds at the node u^t and the number $|\mathcal{S}_t|$.

A *witness* at node v_t is a time-like sequence of edges $\mathcal{S}_t$ such that $|\mathcal{S}| \geq t - n_v(t)$. In Lemma 6.9, we will demonstrate a witness for v_t for all $v \in \mathcal{V}$ and $t \geq 0$. The technique is very similar to the proof of Lemma 6.5 and hence we will only provide a sketch of the proof. After that we will use Lemma 6.8 to prove that $P(t - n_v(t) \geq m) \leq (\Delta + 1)^t \binom{t}{m} 2^{-m\beta'/2}$ for any $v \in \mathcal{V}$.

Lemma 6.9. *If after t rounds of communication, node v has performed $n_v(t)$ iterations of the protocol, then there exists a time-like sequence, $\mathcal{S}_t$ of t edges in $\vec{\mathcal{E}}_{\mathcal{T}}$ ending in node v_t with $|\mathcal{S}_t| > t - n_v(t)$*

Proof. The proof is obtained by repeating the same argument as in the proof of Lemma 6.5 with the word 'erasure' replaced with the word 'tree code error'. The only case that needs a little bit of clarification is case 2-b-ii, i.e., round t is wasted at node v and $B \cap Z \neq \emptyset$, where B and Z retain the same meaning as before. In

this case, as before, there is a neighbor $u \in \mathcal{N}_v$ such that $n_{vu}(t) = n_u(t-1) - 2$. From Algorithm 5, it is clear that node the information $x^u_{n_u(t-1)-1}$ was encoded and transmitted by node u to node v in round $t-1$ or before. Therefore, the error interval on the edge $(v_t, u_{t-1}) \in \vec{\mathcal{E}}_{\mathcal{T}}$ contains the interval $[t-1, t]$. Let the witness at node u_{t-1} be $\mathcal{S}_{t-1,u}$. Append the edge (v_t, u_{t-1}) to $\mathcal{S}_{t-1,u}$ to get a new time-like sequence which we call $\mathcal{S}_{t,v}$. We claim that $\mathcal{S}_{t,v}$ is a witness at v_t. This proof of this claim follows from the following observations

1. When applying Lemma 6.8, we only to care about error intervals on the same edge at different times

2. The edge (v, u) appears in the time-like sequence $\mathcal{S}_{t,v}$ for round t and hence, it can possibly appear again only in $\mathcal{S}_{t,v}$ in round $t-2$ or earlier. So, the length of the union of the error intervals on the edges $(v_\tau, u_{\tau-1}) \in \mathcal{S}_{t-1,u}$ increases by at least 2 with the addition of the edge (v_t, u_{t-1}). Hence we have

$$|\mathcal{S}_{t,v}| \geq |\mathcal{S}_{t-1,u}| + 2 \geq t - 1 - n_u(t-1) = t - n_v(t)$$

   This completes the proof.

Putting together Lemma 6.9 and Lemma 6.8, we have

$$P(t - n_v(t) \geq m) \leq (\Delta + 1)^t \binom{t}{m} 2^{-\beta' m/2}$$

This completes the proof. □

6.6.3 Proof of Theorem 6.2

The bound $(1-p)^{|\mathcal{E}|}$ is intuitively motivated by the following observation, in a given round of communication, $(1-p)^{|\mathcal{E}|}$ is the probability that none of the edges are erased. As a result one would expect the fraction of communication rounds in which nodes can perform an iteration of the protocol to be approximately $(1-p)^{|\mathcal{E}|}$. The above observation alone would not render a proof because successful communication could

also mean that a node received only 'waits' from its neighbors and hence could not compute an iteration of the protocol. The proof idea is simple but conveying it requires some setup. Let $W_{uv}^{(t)}$ denote the event where node v transmits a 'wait' to node u in round t. We introduce the following definition

Definition 6.3. *Consider nodes v, u, u' such that $u \in \mathcal{N}_v$ and $u' \in \mathcal{N}_u$. Also suppose that node v transmits a 'wait' to node u in round τ and node u transmits a 'wait' to node u' in round $\tau + 1$, i.e., events $W_{uv}^{(\tau)}$ and $W_{u'u}^{(\tau+1)}$ happen.* Then $W_{uv}^{(\tau)}$ is said to have caused $W_{u'u}^{(\tau+1)}$ *if both the following conditions hold*

(a) $n_u(\tau - 1) = 1 + n_{uv}(\tau - 1)$

(b) $n_{u'u}(\tau) = n_u(\tau)$

To understand the definition, observe that condition (a) implies that node v is a bottleneck node for node u in round τ and condition (b) implies that node u' already knows $n_u(\tau)$ after round τ. Node u could not perform a new iteration in round τ since it received a 'wait' from a bottleneck node (in this case v) and hence sent a 'wait' to node u'. So, it is natural to blame $W_{uv}^{(\tau)}$ for $W_{u'u}^{(\tau+1)}$. Note that Definition 6.3 is further justified by the observation that a 'wait' in round τ will either have an effect in round $\tau + 1$ or will never. Also note that Definition 6.3 can be extended to more than two waits by having conditions (a) and (b) hold for every pair of successive 'wait' events.

With that, we are now ready to state the main lemma. The lemma essentially implies that 'waits' do not loop in the network. In other words, if in round τ a node v transmits a 'wait', then this 'wait' will not *cause* the same node v to transmit another 'wait' in a future round $\tau' > \tau$.

Lemma 6.10 ('Waits' do not loop)**.** *Consider the sequence of events $\{W_{u_{i+1}u_i}^{(\tau+i-1)}\}_{i=1}^{\ell}$ such that $W_{u_{i+1}u_i}^{(\tau+i-1)}$ is caused by $W_{u_iu_{i-1}}^{(\tau+i-2)}$ for all $2 \leq i \leq \ell$. Then the nodes $\{u_i\}_{i=1}^{\ell}$ are all distinct.*

Proof. Node u_1 sent a 'wait' to node u_2 in round τ implies that $n_{u_1}(\tau-1) = n_{u_2u_1}(\tau-1)$. Furthermore, since $W_{u_3u_2}^{(\tau+1)}$ is caused by $W_{u_2u_1}^{(\tau)}$, conditions (a) and (b) in Definition

6.3 apply. In particular, condition (a) together with the first observation gives $n_{u_2}(\tau-1) = n_{u_1}(\tau-1)+1$. Since node u_2 could not perform a new iteration of the protocol, we have $n_{u_2}(\tau) = n_{u_2}(\tau-1) = n_{u_1}(\tau-1)+1$. Repeating this argument for the remaining nodes, we get

$$n_{u_{i+1}}(\tau+i-1) = n_{u_i}(\tau_i-2),\ \forall\ 1 \le i \le \ell \tag{6.24}$$

Now suppose the nodes $\{u_i\}_{i=1}^{\ell}$ are not all distinct. In particular, suppose $u_\ell = u_1$. Then from (6.24), we have $n_{u_1}(\tau+\ell-2) = n_{u_\ell}(\tau+\ell-2) = \ell-1+n_{u_1}(\tau-1)$ which is not possible since $n_{u_1}(.)$ can increment by at most 1 in each round and $n_{u_1}(\tau) = n_{u_1}(\tau-1)$.

One will similarly arrive at a contradiction if any other node repeats in $\{u_i\}_{i=1}^{\ell}$. □

The implication of Lemma 6.10 is clear. If a node v sends a 'wait' in round τ to any of its neighbors, then this 'wait' will not by itself stop node v from performing an iteration of the protocol in a future round.

We are now ready to provide the main argument. Let $d(v,u)$ denote the length of the shortest path from node u to node v. So, if $v \in \mathcal{N}_u$, then $d(v,u) = 1$ and $d(v,v) = 0$. Let the diameter of the graph be δ, i.e., $\delta = \max_{u,v\in\mathcal{V}} d(v,u)$. And for an edge $e_{uu'} \equiv (u,u') \in \mathcal{E}$, we define

$$d(v, e_{uu'}) = \min\{d(v,u), d(v,u')\}$$

Let $\mathcal{E}_v^{(i)} \triangleq \{e \in \mathcal{E} \mid d(v,e) = i\}$. In view of Lemma 6.10, it is not difficult to see that an erasure on an edge in $\mathcal{E}_v^{(i)}$ in round τ will have an effect (if any) at node v only in round $\tau+i$. Let $A_{i,\tau}$ denote the event that there is an erasure on an edge in $\mathcal{E}_v^{(i)}$ in round τ. Then for $\tau \ge \delta$, it is easy to see that $\cap_{i=0}^{\delta} A_{i,\tau-i}^c$ implies that the round τ at node v is *not* wasted, i.e., node v can compute an iteration of the protocol. In other words

$$P(n_v(\tau) = n_v(\tau-1)) \le 1 - P\left(\bigcap_{i=0}^{\delta} A_{i,\tau-i}^c\right) = 1-(1-p)^{|\mathcal{E}|}$$

Due to the erasure model, note that the even $A_{i,\tau}$ is independent of $A_{i',\tau'}$ for $(i,\tau) \neq (i',\tau')$. Let

$$X_\tau = \mathbb{1}_{[n_v(\tau)=n_v(\tau-1)]}, \;\; Y_\tau = \mathbb{1}_{\left[\cup_{i=0}^{\delta} A_{i,\tau-i}\right]}$$

Then from the above argument $X_\tau = 1$ implies $Y_\tau = 1$ and $\{Y_\tau\}$ are independent Bernoulli random variables. Note that $P(Y_\tau = 1) \leq 1 - (1-p)^{|\mathcal{E}|}$. Let $R' = \frac{n_v(t)}{t}$, then we have

$$\begin{aligned} P(t - n_v(t) = m) = P\left(\sum_{\tau=0}^{t} X_\tau = m\right) \leq P\left(\sum_{\tau=0}^{t} Y_\tau \geq m\right) &\leq 2^{-tD(1-R',1-(1-p)^{|\mathcal{E}|})} \\ &= 2^{-tD(R',(1-p)^{|\mathcal{E}|})} \end{aligned}$$

The last inequality follows from a standard Chernoff bounding technique and is true whenever $R' < (1-p)^{|\mathcal{E}|}$. Union bounding over all nodes $v \in \mathcal{V}$, we have

$$P\left(\exists\, v \in \mathcal{V} \;\ni n_v(t) \leq R't\right) \leq N 2^{-tD(R',(1-p)^{|\mathcal{E}|})}$$

This completes the proof.

Chapter 7

Application to Consensus Over Erasure Channels

7.1 Introduction

In a network of agents, consensus refers to the process of achieving agreement between the agents in a distributed manner. Consensus problems, and in particular the problem of reaching consensus on the average of the values of the agents, have been around for a while and are often used to serve as a test case for studying distributed computation and decision making between a group of nodes/processors/dynamical systems [46, 73, 74, 102, 103, 107]. Most of the work in this area assumes that the agents are connected via a fixed underlying graph or network. In many applications, however, the links in the underlying graph are noisy or unreliable. In the context of consensus problems, the unreliability of communication links between nodes has been traditionally modeled by allowing the underlying graph to vary with time. In other words, at each time instant some of the links are allowed to be erased, and depending on the realization of the link erasures, the underlying graph at each time instant is assumed to be a subgraph of the original graph. Furthermore, the distributed algorithm for reaching consensus remains unchanged: the same distributed averaging algorithm is used, except that only the information received at each time is used. An important assumption that is implicitly made in this model is that the erasures are symmetric: if at time t the packet from node i to node j is dropped, the

same is true for the packet transmitted from node j to node i. In practical wireless communication systems this assumption is patently unreasonable: the additive noise at the two nodes are independent and, furthermore, communication in the two directions occurs at either different times or over different frequency bands. If standard averaging protocols are performed, this loss of symmetry can prohibit the network from reaching consensus to the true average (standard consensus protocols require that the "update" matrix be doubly stochastic, something that cannot be guaranteed in the asymmetric case).

The goal of this Chapter is to explore the use of channel coding to improve the performance of consensus algorithms, especially in the asymmetric case. For asymmetric erasures we show that tree codes can be used to simulate the performance of the original *unerased* graph. Thus, unlike conventional consensus methods, we can *guarantee* convergence to the average in the asymmetric case. As expected, the price is a slowdown in the convergence rate, relative to the convergence rate of the unerased network. Nonetheless, the slowdown is still often faster than the convergence rate of conventional consensus algorithms over erasure links.

7.2 Background

For a fixed communication graph $\mathcal{G}$, a typical algorithm to achieve consensus is of the following form.

$$x_{k+1}^i = w_{ii} x_k^i + \sum_j w_{ij} x_k^j \tag{7.1}$$

W obeys the underlying graph, i.e., for $i \neq j$, $W_{ij} = 0$ if $(i,j) \notin \mathcal{E}$. In other words, each node updates its value by taking a weighted sum of its own previous value with those of its neighbors. In short, the equation can be written as

$$x_{k+1} = W x_k \tag{7.2}$$

Such an algorithm is said to achieve consensus if

$$\lim_{k\to\infty} x_k^i = \overline{x}_0 \triangleq \frac{1}{N}\sum_j x_0^j \tag{7.3}$$

In such a static setup where the weights and the underlying interaction graph does not change with time, it is clear that consensus is achieved if and only if

$$\lim_{k\to\infty} W^k = \frac{1}{N}\mathbf{1}\mathbf{1}^T \tag{7.4}$$

Further (7.4) holds if and only if the following conditions hold (e.g., [112])

1. W is doubly stochastic, i.e.,

$$\mathbf{1}^T W = \mathbf{1}^T,\ \ W\mathbf{1} = \mathbf{1} \tag{7.5}$$

2. $\rho\left(W - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) < 1$

where $\rho(.)$ is the spectral radius. Note that $x_k = W^k x_0$. Under the above conditions, $x_k \to \frac{1}{N}\mathbf{1}\mathbf{1}^T x_o = \overline{x}_0 ones$. The convergence rate, $\mu(W)$, of the above consensus algorithm is formally defined as

$$\mu(W) = \sup_{x_o \neq \overline{x}_0\mathbf{1}} \lim_{k\to\infty} \left[\frac{\|x_k - \overline{x}_0\mathbf{1}\|}{\|x_o - \overline{x}_0\mathbf{1}\|}\right]^{\frac{1}{k}} \tag{7.6}$$

and is given by $\mu(W) = \rho\left(W - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)$. There is a considerable amount of work that explores different choices of W and how it affects the rate of convergence of the consensus algorithm (e.g., [112]).

For ease of exposition, we use a specific but natural choice of W (e.g., [74]) given by $W = I - \epsilon\mathcal{L}$, where $\mathcal{L}$ is the Laplacian of the interaction graph $\mathcal{G}$, i.e., $\mathcal{L} = D - A$. $D = \text{diag}\{\Delta_i\}$ where Δ_i is the degree of node i. Let $0 = \lambda_N(\mathcal{L}) \leq \lambda_{N-1}(\mathcal{L}) \leq \ldots \leq \lambda_1(\mathcal{L})$ denote the eigenvalues of $\mathcal{L}$. The multiplicity of the zero eigen value is the number of connected components in the graph and $\lambda_{N-1}(\mathcal{L}) > 0$ if and only if the graph is connected.

Table 7.1: Notation for Chapter 7

x_0^i	the initial value at node i
x_0	column of x_0^i's
$\overline{x}_0$	the initial average, i.e., $\frac{1}{N}\mathbf{1}^T x_0$
$\rho(.)$	spectral radius of a matrix
$A \circ B$	Hadamard product,i.e., $(A \circ B)_{ij} = A_{ij}B_{ij}$
$A \otimes B$	Kronecker product

For such a choice of W, the spectral radius is given by $\rho(W - \frac{1}{N}\mathbf{1}\mathbf{1}^T) = \max\{1 - \epsilon\lambda_{N-1}(\mathcal{L}), \epsilon\lambda_1(\mathcal{L}) - 1\}$. We state this as a lemma for later reference.

Lemma 7.1. *The convergence rate, μ, of* (7.1) *with $W = I - \epsilon\mathcal{L}$ is*

$$\mu = \max\{1 - \epsilon\lambda_{N-1}(\mathcal{L}), \epsilon\lambda_1(\mathcal{L}) - 1\} \tag{7.7}$$

So, the conditions 1) and 2) above are satisfied if and only if $\epsilon < \frac{2}{\lambda_1(\mathcal{L})}$. Furthermore, the convergence rate μ is maximized when the two quantities in (7.7) coincide, i.e., when

$$\epsilon = \epsilon^* = \frac{2}{\lambda_1(\mathcal{L}) + \lambda_{N-1}(\mathcal{L})} \tag{7.8}$$

In particular, any $\epsilon < 1/\Delta$ will work where $\Delta = \max_i \Delta_i$. We remark that the results presented here are independent of the choice of the weight matrix W. Whenever we wish to write closed form expressions for the convergence rates, we use the specific choice $W = I - \epsilon^*\mathcal{L}$ for simplicity.

Note that $a_{ii} = 0$. Let x_0^i denote the initial value at node i. The objective is for the nodes to compute the global average $r = \frac{1}{N}\mathbf{1}^T x_0$, where $\mathbf{1}$ denotes an N-dimensional column of ones and x_0 is the column vector of the x_0^i's.

7.2.1 Noisy Links

In practice, the communication links between nodes can be unreliable. Conventionally, this has been taken into account by allowing the interaction topology to change with time. So, at time k, the connectivity between nodes is described by the graph $\mathcal{G}_k$ where $\mathcal{G}_k$ can now vary with time. There is a considerable amount of literature on the problem of achieving consensus under such time varying interaction topologies [17, 41, 69, 73, 107].

The literature on consensus over time varying topologies only captures the symmetric case. Even though, consensus under very general conditions has been established, not much appears to be available by way of the rate of convergence. Under the asymmetric erasure model, the resulting interaction graph is effectively directed. An edge between node i and j is replaced by a pair of directed edges. The effective graph at any time depends on the packets that were erased in that round. Under this setup, we define the adjacency matrix $A = [a_{ij}]$ and the Laplacian $\mathcal{L}$ as follows; $a_{ij} = 1$ if $(i \leftarrow j) \in \mathcal{E}$ and $\mathcal{L} = D - A$ with $D = \text{diag}\{\Delta_i\}$ and $\Delta_i = \sum_j a_{ij}$. The resulting adjacency matrix and the Laplacian are not symmetric in general. As a result, they are not doubly stochastic either, i.e., $\mathbf{1}^T\mathcal{L} \neq \mathbf{1}^T$. When the graph $\mathcal{G}$ is directed, (Olfati-Saber Murray 2007) prove that average consensus is achieved using a fixed $W = I - \epsilon\mathcal{L}$ if and only if the interaction graph $\mathcal{G}$ is balanced, i.e., the in-degree of each node is equal to its out degree [73]. But when the link failures are random, the resulting interaction graph will generally not be balanced at every time step and average consensus cannot be achieved.

Achieving average consensus can be naturally viewed as an instance of interactive protocols over graphs. So we can simulate it over noisy links using tree codes as described in Algorithm 5 in Chapter 6.

Before proceeding further, it is important to note that the simulation algorithm of Chapter 6 is universal. Concomitant with this universality is that it may be an overkill for specific instances of interactive protocols such as averaging. While Algorithm 5 will exactly simulate every iteration of the average consensus protocol, this may not

be necessary in order to achieve average consensus. It is conceivable that a simulation algorithm that allows mistakes will achieve average consensus faster than a universal algorithm such as Algorithm 5. This is not a focus of the current Chapter.

7.3 Coding Vs. No Coding

When there are erasures and when there is no coding, an iteration of the consensus algorithm at node i is given by

$$x_{k+1}^i = x_k^i - \epsilon \sum_j a_{ij} X_k^{ij} (x_k^i - x_k^j) \tag{7.9}$$

The effective adjacency matrix at time k is then $A_k = A \circ X_k$, where $X_k = [X_k^{ij}]$. The associated Laplacian is $\mathcal{L}_k = D_k - A_k$ where $D_k^i = \sum_j A_k^{ij} = \sum_j a_{ij} X_k^{ij}$.

To study the effect of coding we need to distinguish between the symmetric and asymmetric erasure models. When the erasures are symmetric, i.e., when $X_k^{ij} = X_k^{ji}$, this means that node i (respectively, node j) *knows* what node j (respectively, i) has received. For example, if node i successfully received a packet from node j, it knows that node j also successfully received the packet intended for it; alternately if node i receives an erasure from node j, it knows that the packet intended for node j was also erased. In this case, the links between the different nodes are erasure links with feedback (where the transmitter knows what the receiver receives). For erasure links with feedback the optimal coding scheme on each link is *retransmission*, i.e., the transmitter retransmits its packet until it is received at the receiver. When the erasures are not symmetric, one needs the more sophisticated tree codes.

7.3.1 Symmetric Erasures

The recursion (7.9) can written as $x_{k+1} = (I - \epsilon \mathcal{L}_k) x_k$. The convergence rate of this recursion when erasures are symmetric is given by the following lemma.

Lemma 7.2 (Symmetric Erasures)**.** *When the erasures are symmetric and i.i.d over*

time and space, the convergence rate of (7.9), $\mu_{\overline{c}}^s$ *which we define as*

$$\mu_{\overline{c}}^s = \sup_{x_o \neq \overline{x}_0 \mathbf{1}} \lim_{k\to\infty} \left[\frac{\mathbb{E}\|x_k - \overline{x}_0 \mathbf{1}\|^2}{\|x_o - \overline{x}_0 \mathbf{1}\|^2} \right]^{\frac{1}{2k}} \tag{7.10}$$

is given by

$$\mu_{\overline{c}}^s = \sqrt{\lambda_2(\Gamma_s)} \tag{7.11}$$

where $\Gamma_s = \mathbb{E}(I - \epsilon\mathcal{L}_0) \otimes (I - \epsilon\mathcal{L}_0)$ *is a deterministic matrix that is a function of* $\epsilon, p, \mathcal{L}$ *and can be computed explicitly in closed form. The subscript* $\overline{c}$ *indicates that there is no coding and the subscript* s *in* Γ_s *is because the erasures are symmetric*

Proof. See Appendix 7.4.1. □

In this case, note that even without coding, the nodes achieve average consensus albeit at a slower rate depending on the erasure probability p.

Now consider using repetition coding. To understand the rationale behind even considering repetition coding, recall that the recursion (7.9) can be written as $x_{k+1} = (I - \epsilon\mathcal{L}_k)x_k$. Take expectation on both sides to get $\overline{x}_{k+1} = (I - \epsilon\overline{\mathcal{L}})\overline{x}_k$ where the bar indicates that they are expected values[1]. Since the link erasure probability is p, $\overline{\mathcal{L}} = I - \epsilon(1-p)\mathcal{L}$. Suppose $\epsilon = \epsilon^*$ as in (7.8). Then using Lemma 7.1, the rate of convergence of $\overline{x}_k$ to $\overline{x}_0\mathbf{1}$ can be calculated as

$$\begin{aligned}
\overline{\mu} &= \max\left\{1 - \epsilon\lambda_{N-1}(\overline{\mathcal{L}}), \epsilon\lambda_1(\overline{\mathcal{L}}) - 1\right\} && (7.12)\\
&= \max\left\{1 - \epsilon(1-p)\lambda_{N-1}(\mathcal{L}), \epsilon(1-p)\lambda_1(\mathcal{L}) - 1\right\} && (7.13)\\
&= \frac{\lambda_1(\mathcal{L}) - \lambda_{N-1}(\mathcal{L})}{\lambda_1(\mathcal{L}) + \lambda_{N-1}(\mathcal{L})} + \frac{2p\lambda_{N-1}}{\lambda_1(\mathcal{L}) + \lambda_{N-1}(\mathcal{L})} && (7.14)\\
&= \mu + \frac{2p\lambda_{N-1}}{\lambda_1(\mathcal{L}) + \lambda_{N-1}(\mathcal{L})} && (7.15)
\end{aligned}$$

where μ is the rate of convergence of the consensus protocol on the unerased graph. Clearly $\overline{\mu} > \mu$. Moreover the rate of convergence of x_k to $\overline{x}_0\mathbf{1}$ is even slower (as

[1]this is inconsistent with the notation $\overline{x}_0$ which is a deterministic scalar but should not cause any confusion

compared to $\overline{x}_k$ converging to $\overline{x}_0\mathbf{1}$). Since the repetition code simulates consensus over the unerased graph whose convergence rate is μ, it can potentially result in faster convergence if the overhead due to repetition is not too high.

Using Theorems 6.1 and 6.2, we can determine the convergence rate, μ_c^s, of consensus using the repetition code (i.e., Algorithm 4) and it is given by

$$\mu_c^s \leq \min\{\mu^{R(p)}, \mu^{(1-p)|\mathcal{E}|}\} \tag{7.16}$$

μ is defined in (7.7). The superscript and subscript in μ_c^s denote that it is the convergence rate with coding under symmetric erasures. So, whenever $\mu_c^s < \mu_{\overline{c}}^s$, coding offers an advantage. In practice though, the computational overhead of doing repetition coding would probably far outweigh any benefits of being able to reach consensus faster. The more interesting and relevant case is when erasures are asymmetric in which there is no recourse coding.

7.3.2 Asymmetric Erasures

Since X_k^{ij} and X_k^{ji} are independent, they are not equal in general. Note that $\mathcal{L}_k\mathbf{1} = \mathbf{1}$ but $\mathbf{1}^T\mathcal{L}_k \neq \mathbf{1}^T$ in general which violates (7.5). Furthermore, the associated graph is not balanced[2] either, i.e., $\sum_j a_{ij}X_k^{ij} \neq \sum_i a_{ji}X_k^{ji}$, in general. In this case, the nodes will not achieve *average* consensus. But under very mild conditions, it is well known that the nodes achieve an agreement, i.e., $x_k \to Y\mathbf{1}$ where Y is a random variable that does not necessarily concentrate around the initial average r. Nevertheless the nodes reach agreement and we will characterize the rate of convergence below. But tree codes allow us to simulate the original recursions, i.e., (7.1), and hence guarantee asymptotic average consensus. Here, we characterize the mean-squared error of the state from average consensus when no error correction is used.

Lemma 7.3 (Asymmetric Erasures)**.** *When the erasures are asymmetric and i.i.d*

[2]A graph is said to be balanced if for every node in-degree is equal to out-degree.

over time and space, we have

$$\mathbb{E}\|x_k - \overline{x}_0\mathbf{1}\|^2 = (x_o - \overline{x}_0\mathbf{1})^T \otimes (x_o - \overline{x}_0\mathbf{1})^T \Gamma_a^k vec(I) \tag{7.17}$$

Here I is an $N \times N$ identity matrix and

$$\Gamma_a = \mathbb{E}(I - \epsilon\mathcal{L}_0^T) \otimes (I - \epsilon\mathcal{L}_0^T) \tag{7.18}$$

where Γ_a is a deterministic matrix that is a function of $\epsilon, p, \mathcal{L}$ and can be computed explicitly in closed form. Furthermore $\rho(\Gamma_a) = 1$.

Proof. See Appendix 7.4.2. □

Note that $\mathbf{1}^T\Gamma_a = \mathbf{1}^T$ but $\Gamma_a\mathbf{1} \neq \mathbf{1}$. Let c, $\|c\| = 1$ be the right eigen vector of Γ_a corresponding to eigen value 1, i.e., $\Gamma_a c = c$. Then, it is easy to see that $\lim_{k\to\infty} \Gamma_a^k = \frac{1}{N} c\mathbf{1}^T$. Using this in (7.17), we get

$$\lim_{k\to\infty} \mathbb{E}\|x_k - \overline{x}_0\mathbf{1}\|^2 = (x_o - \overline{x}_0\mathbf{1})^T \otimes (x_o - \overline{x}_0\mathbf{1})^T c \tag{7.19}$$

This proves that one cannot achieve average consensus without coding when link failures are asymmetric. So, a major benefit of using tree codes in such cases is to *guarantee* average consensus. Note that we ignore quantization effects which is justified by the packet sizes used in practice. We can easily compute the rate of convergence of the consensus protocol when tree codes are used. Recall that the overall rate of the simulation protocol (i.e., Algorithm 5) is at least $r\rho(r,p)$ where r is the rate of the tree code, p is the probability of packet erasure and $\rho(r,p)$ is defined in (6.13) as

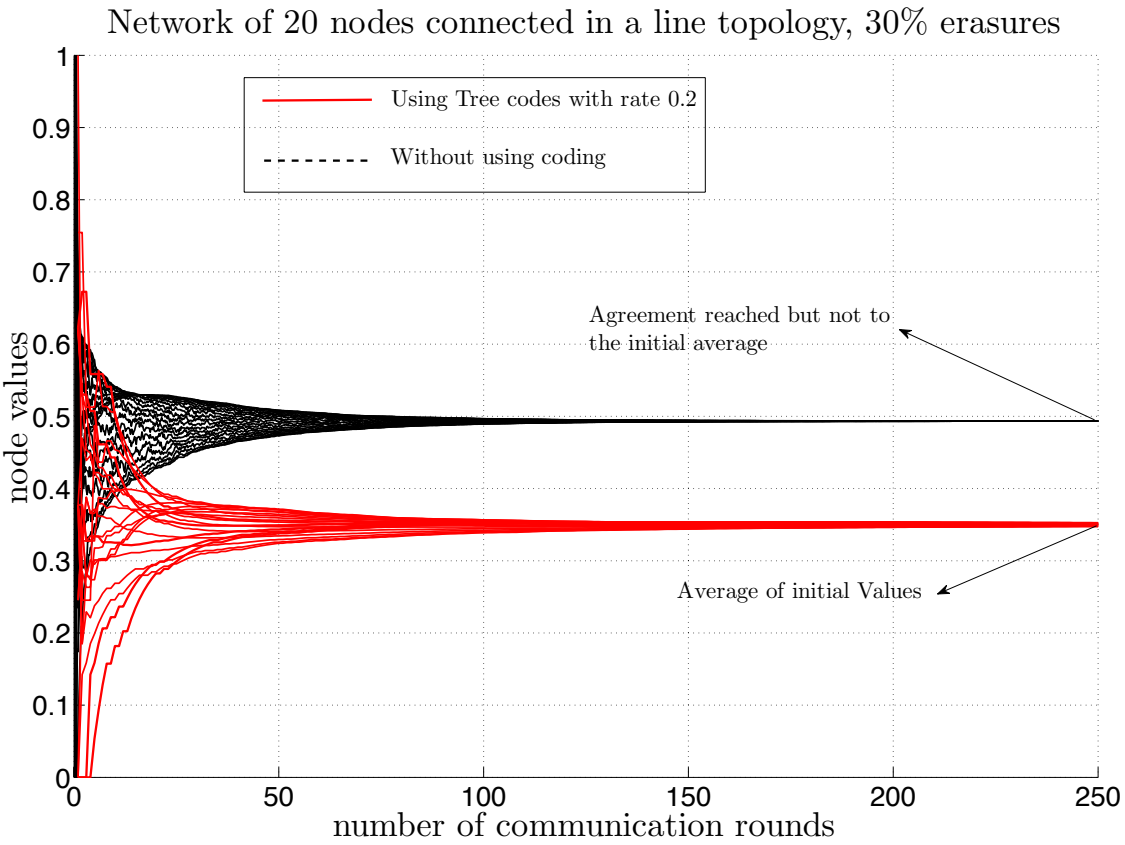
$$\rho(r,p) \triangleq \sup_{R' \geq 0} \{R' \mid (1 - R')\beta'/2 > H(R') + \log(\Delta + 1)\}$$

The effective rate of convergence to average consensus achieved by using tree codes is no worse than $\mu^{r\rho(r,p)}$).

7.3.3 A Simulation

We will perform a simple simulation to demonstrate the effectiveness of tree codes in achieving average consensus and achieving it quickly. We will use a graph of 20 nodes connected in a straight line as depicted in Fig 7.3.3. The packet length is 16. When there is no coding, nodes exchange one packet each in every communication round. For coding, we generate a random code from the Toeplitz ensemble (e.g., Chapter 4) with rate 1/5, i.e., every packet is mapped to 5 packets and a communication round now consists of exchanging these 5 packets between every pair of rounds. Each node is initialized randomly with 0 or 1. Sample trajectories of the values at every node in the graph are plotted in Figure 7.3.3. The plot clearly illustrates the fact that nodes do not achieve average consensus without coding while they do with tree codes.

(a) A line graph with 20 nodes



(b) Tree codes achieve average consensus. The slow down due to coding is visible in the plot.

Figure 7.1: One needs coding to achieve average consensus when packet erasures are asymmetric

7.4 Appendices

7.4.1 Proof of Lemma 7.2

Note that $\mathcal{L}_k\mathbf{1} = 0$ whether or not the erasures are symmetric. Recall that $r = \frac{1}{N}\mathbf{1}^T x_0$.

$$x_k - \overline{x}_0\mathbf{1} = (I - \epsilon\mathcal{L}_{k-1})(x_{k-1} - \overline{x}_0\mathbf{1}) \tag{7.20a}$$

$$x_k - \overline{x}_0\mathbf{1} = \tag{7.20b}$$

$$\underbrace{(I - \epsilon\mathcal{L}_{k-1})(I - \epsilon\mathcal{L}_{k-2})\dots(I - \epsilon\mathcal{L}_0)}_{\triangleq Y_k}(x_0 - \overline{x}_0\mathbf{1}) \tag{7.20c}$$

$$\begin{aligned}\mathbb{E}\|x_k - \overline{x}_0\mathbf{1}\|^2 &= (x_0 - \overline{x}_0\mathbf{1})^T\mathbb{E}Y_k^T Y_k(x_0 - \overline{x}_0\mathbf{1}) \\ &= (x_0 - \overline{x}_0\mathbf{1})^T \otimes (x_0 - \overline{x}_0\mathbf{1})^T vec(P_k)\end{aligned} \tag{7.21}$$

where $P_k = \mathbb{E}Y_k^T Y_k$. Recall that the erasure process is independent over time and across links. Then we have

$$P_k = \mathbb{E}(I - \epsilon\mathcal{L}_0^T)P_{k-1}(I - \epsilon\mathcal{L}_0) \tag{7.22a}$$

$$vec(P_k) = \Gamma_s vec(P_{k-1}), \text{ where} \tag{7.22b}$$

$$\Gamma_s = \mathbb{E}(I - \epsilon\mathcal{L}_0^T) \otimes (I - \epsilon\mathcal{L}_0^T) \tag{7.22c}$$

Since erasures are symmetric, $\mathcal{L}_0^T = \mathcal{L}_0$. Furthermore, we have $vec(P_k) = \Gamma_s^k vec(I)$, where I is an $N \times N$ identity matrix. Putting (7.21) and (7.22) together, we get

$$\mathbb{E}\|x_k - \overline{x}_0\mathbf{1}\|^2 = (x_0 - \overline{x}_0\mathbf{1})^T \otimes (x_0 - \overline{x}_0\mathbf{1})^T\Gamma_s^k vec(I) \tag{7.23}$$

So, the rate of convergence of the consensus algorithm in the absence of coding is clearly determined by Γ_s. Observe that Γ_s is doubly stochastic, i.e., $\mathbf{1}^T\Gamma_s = \mathbf{1}^T$ and $\Gamma_s\mathbf{1} = \mathbf{1}$. It has one eigen value at 1 and all others are strictly smaller than 1 in magnitude. Let $\lambda_2(\Gamma_s)$ denote the second largest eigen value in magnitude. Then

clearly

$$\lim_{k\to\infty} \Gamma_s^k = \frac{1}{N^2}\mathbf{1}\mathbf{1}^T \tag{7.24}$$

and the rate of convergence is given by $\mu_{\bar{c}}^s = \sqrt{\lambda_2(\Gamma_s)}$

7.4.2 Proof of Lemma 7.3

Except the claim $\rho(\Gamma_a) = 1$, everything else follows from Appendix 7.4.1. Since $\Gamma_a = \mathbb{E}(I - \epsilon\mathcal{L}_0^T) \otimes (I - \epsilon\mathcal{L}_0^T)$, the claim $\rho(\Gamma_a) = 1$ follows if $\rho(I - \epsilon\mathcal{L}_0) = 1$ which is what we show. Recall that the random variable X_0^{ij} is defined as $X_0^{ij} = 0$ if the link $j \to i$ is erased at time 0 and $X_0^{ij} = 1$ otherwise. For brevity, we will write X^{ij} instead of X_0^{ij}. Then it is easy to verify that one can write $\mathcal{L}_0$ as follows

$$\mathcal{L}_0 = \sum a_{ij} X^{ij} e_i (e_i - e_j)^T \tag{7.25}$$

where e_i is the i^{th} unit vector. In particular, the underlying Laplacian in the absence of any erasures can be written as $\mathcal{L} = \sum a_{ij} e_i (e_i - e_j)^T$. For any $x \in \mathbb{R}^N$, we have

$$\begin{aligned} x^T(I - \epsilon\mathcal{L}_0)x &= x^T \left(I - \frac{\epsilon}{2}(\mathcal{L}_0 + \mathcal{L}_0^T)\right) x \\ &= \|x\|^2 - \frac{\epsilon}{2}\sum a_{ij} X^{ij} (x_i - x_j)^2 \leq \|x\|^2 \end{aligned} \tag{7.26}$$

Furthermore,

$$\begin{aligned} \|x\|^2 - \frac{\epsilon}{2}\sum a_{ij} X^{ij} (x_i - x_j)^2 &\geq \|x\|^2 - \frac{\epsilon}{2}\sum a_{ij} (x_i - x_j)^2 \\ &= x^T(I - \epsilon\mathcal{L})x \geq -\|x\|^2 \end{aligned} \tag{7.27}$$

The last inequality follows from the fact that $\rho(I - \epsilon\mathcal{L}) = 1$. Combining (7.26) and (7.27), we have $|x^T(I - \epsilon\mathcal{L}_0)x| \leq \|x\|^2$ for all $x \in \mathbb{R}^N$ which implies that $\rho(I - \epsilon\mathcal{L}_0) \leq 1$. But $\mathcal{L}_0\mathbf{1} = \mathbf{1}$, so $\rho(I - \epsilon\mathcal{L}_0) = 1$. Therefore $\rho(\Gamma_a) = 1$. This completes the proof.

Chapter 8

Conclusions and Future Directions

8.1 Conclusions

Fueled by rapid advances in embedded systems technology and communications infrastructure, cheaply available *smart* devices with small form factors, capable of sensing, computing and wireless communications, have proliferated throughout many applications. These advances have enabled monitoring and data collection from an unprecedented variety of areas encompassing weather and environment, medical care, energy consumption, vehicular traffic, public spaces, structural health monitoring of man-made constructions and even online social networks. The next logical step in this evolution is to use this data to control and influence the physical world in an automated manner with minimal human intervention. Possible instances of this new paradigm include the smart grid, fully autonomous highway systems, and networked city services just to mention a few. Widely referred to as cyberphysical systems and/or networked control systems, they are conjectured to have a complexity comparable to that of biological systems.

Essential to understaning and realizing cyberphysical systems in practice is an integrated systems theory of computing, communications and control. There has been significant effort by the research community in this direction in recent years [1, 44, 57, 70, 84]. Two important features of networked control systems are decentralization of information and the need to exchange it over potentially unreliable communication networks. Consequently, one of the key challenges (e.g., [70]) in building

future networked control systems is to integrate information theory and control theory, two fields that have traditionally developed almost completely independently of each other. The work presented in this thesis is motivated by this challenge.

In the first part of the thesis, we focused on decentralized estimation in the context of sensor networks. This was motivated by applications where sensor communication is subject to severe power and bandwidth constraints. We proposed a novel particle filtering technique called *Kalman-like particle filter (KLPF)* for optimally tracking a linear Gaussian state-space process using quantized measurements. We showed through simulations that the proposed filter outperforms conventional particle filtering techniques by orders of magnitude. Furthermore, unlike conventional numerical techniques, the operations performed in the KLPF converge to the regular Kalman filter as the quantization becomes finer. The KLPF constitutes an efficient approach to peform optimal LQG control using quantized measurements. In this setup, we assumed that the communication between the sensors and controller is noiseless although it is rate limited. But the situation is different if the communication is stochastic and noisy and this is the subject of the second part of the thesis.

Conventional information theoretic techniques achieve communication reliability at the expense of encoding and decoding *delay*. Larger the delay, higher the reliability. Control theory on the other hand deals with *real-time* constraints. Delay in the feedback loop can lead to severe loss of performance and/or instability. As a result, when dealing with networked control systems that have noisy communication channels in their feedback loop, one has to rethink how to achieve communication reliability in a way that is compatible with control objectives.

To address such a scenario, through the early and late 1990's, a new information theoretic notion called *anytime reliability* and a new coding paradigm called *tree codes* was proposed in [84] and [92] respectively. Tree codes are central to several distributed applications including distributed computation and distributed control. But there were no explicit constructions of tree codes since and the subject has remained in the realm of pure theory.

For the first time, we gave an explicit construction of tree codes with efficient

encoding and decoding for a class of communications channels called erasure channels which are used in practice to model links under packetized communication, this includes the internet and wireless links. In the process, we have developed novel non-asymptotic sufficient conditions on the kind of communication reliability required to stabilize control systems that have noisy channels in their feedback loop. We also studied the application of tree codes to interactive protocols between a group of agents connected by a communication graph with erasure links. We further illustrated the benefits of this approach through the example of average consensus.

8.2 Future Directions

Some immediate extensions of the work presented in this thesis are as follows

8.2.1 Going Beyond Stabilization

In the context of distributed control, this thesis focused mainly on communication theoretic aspects of stabilizing unstable plants over noisy channels. This has been achieved by insisting that the channel coder and decoder be anytime reliable. Noting that tree codes are anytime reliable under maximum likelihood decoding, we constructed an explicit ensemble of linear time invariant tree codes and showed how to decode them efficiently over the erasure channel. Recall that anytime reliability is only a sufficient condition for stabilization and does not characterize the overall closed loop performance. In practice, it is essential to go beyond mere stabilization and consider the implications to closed loop performance of the various components such as the source coder (i.e., quantizer), the channel coder (or the joint source-channel coder), the decoder and the control law. We will observe through a simple example why maximum likelihood decoding may be suboptimal when control performance is measured by, say, the second moment of the state. This will serve to emphasize the fact that networked control systems should be viewed in a truly integrated manner rather than just as a sum of its parts.

So we will consider optimally estimating the random walk of example 3.1. Recall from Section 2.6 that the separation principle holds and hence optimal control reduces to optimal estimation. The state x_t in example 3.1 can be written as

$$\begin{aligned} x_t &= \lambda x_{t-1} + w_t \\ \implies x_t &= \lambda^t w_1 + \ldots + \lambda w_{t-1} + w_t \end{aligned} \tag{8.1}$$

The minimum mean squared error (MMSE) estimate of the state given the channel outputs till time t, $\hat{x}_{t|t}^{mmse}$, is the conditional mean and is given by

$$\hat{x}_{t|t}^{mmse} = \lambda^t \mathbb{E} w_1 | z_{0:t} + \ldots + \lambda \mathbb{E} w_{t-1} | z_{0:t} + \mathbb{E} w_t | z_{0:t}$$

where z_τ denotes the channel outputs received during time step τ of (8.1). Since $\{w_\tau\}$ is i.i.d Bernoulli(1/2), the optimal source coder for this problem is obvious, it is to encode each w_t using one bit, say, b_t, i.e., $b_t = 1$ if $w_t = 1$ and $b_t = 0$ otherwise. So

$$\mathbb{E} w_\tau | z_{0:t} = P\left(b_\tau = 1 \,|\, z_{0:t}\right) - P\left(b_\tau = 0 \,|\, z_{0:t}\right)$$

Clearly this is not accomplished by computing the maximum likelihood estimate of $\{b_\tau\}$. One can instead compute $\hat{x}_{t|t}^{mmse}$ using a sequential monte carlo technique such as Algorithm 1. Clearly, maximum likelihood decoding is not necessarily the optimal way to utilize the channel outputs.

The problem of stabilizing unstable plants over noisy channels primarily serves to exemplify the sensitivity of control systems to delay in the feedback loop. An important step is to go beyond mere stabilization and characterize the overall performance and robustness of a decentralized control system. This involves figuring out optimal *real-time* joint source-channel coding/decoding schemes which itself is a major open problem with only basic structural results available.

8.2.2 Anytime reliable codes for other channels

Tree codes require ML decoding to be anytime reliable. Unlike the erasure channel where ML decoding reduces to solving linear equations, ML decoding is computationally intractable for most other channel models. In the case of block coding for example, suboptimal surrogates for ML decoding have been developed for practical use, e.g., message passing algorithms, linear programming decoding and bit flipping for low density parity check codes. Similar analogs of efficiently encodable and decodable constructions of anytime reliable codes do not exist for other channels, e.g., binary symmetric channel and additive white Gaussian noise channel. So, a major open problem is to come up with explicit code constructions with efficient decoding for channels other than the erasure channel. In this context, we explored some causal constructions inspired by low density parity check (LDPC) codes and linear programming decoding. Initial investigations and simulation studies over the binary symmetric channel showed promise. A plausible theoretical roadmap is proposed in Chapter 5 and is an interesting direction to pursue.

8.2.3 Performance of the Kalman-Like Particle filter

Even though the KLPF is an optimal filter, its error performance is not known. The mean-squared error performance is also useful in determining the number of particles that are needed in practice. In general, there are no decentralized estimation algorithms for linear Gaussian state-space processes with provable performance guarantees. The performance of distributed estimation algorithms in the sensor network literature is often predicted based on simplifying assumptions which can sometimes be quite inaccurate. An interesting open problem is to come up with decentralized estimation algorithms for sensor network applications with provable performance guarantees.

Emerging applications of cyberphysical systems provide a fertile ground for many interesting open problems and research directions. The problems listed above constitute only the tip of the iceberg.

Bibliography

[1] D. L. Alderson and J. C. Doyle. Contrasting views of complexity and their implications for network-centric infrastructures. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(4):839 –852, july 2010.

[2] M. S. Amin and B.F. Wollenberg. Toward a smart grid: power delivery for the 21st century. *Power and Energy Magazine, IEEE*, 3, Sept.-Oct. 2005.

[3] C. Andrieu and A. Doucet. Particle filtering for partially observed gaussian state space models. *J. R. Statist. Soc. B*, 64:827–836, 2002.

[4] R. B. Arellano-Valle and M. G. Genton. On fundamental skew distributions. *J. Multivar. Anal.*, 96(1):93–116, 2005.

[5] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55, July 2009.

[6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[7] A. Azzalini and A. Dalla Valle. The multivariate skew-normal distribution. *Biometrika*, 83(4):715–726, 1996.

[8] R. Bansal and T. Başar. Stochastic teams with nonclassical information revisited: When is an affine law optimal? *IEEE Transactions on Automatic Control*, 32, jun 1987.

[9] R. Bansal and T. Baar. Simultaneous design of measurement and control strategies for stochastic systems with feedback. *Automatica*, 25(5):679 – 694, 1989.

[10] Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5), Oct 1974.

[11] A. Barg and G.D. Forney Jr. Random codes: minimum distances and error exponents. *IEEE Transactions on Information Theory*, 48(9):2568 –2573, sep 2002.

[12] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24, May 1978.

[13] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, NY,, 1968.

[14] E. S. Biagioni and K. W. Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16, 2002.

[15] V. S. Borkar and S.K. Mitter. Lqg control with communication constraints. *Communications, computation, control, and signal processing: a tribute to Thomas Kailath*, pages 365–373.

[16] V. S. Borkar, S.K. Mitter, and S.C. Tatikonda. Optimal sequential vector quantization of markov sources. In *Proceedings of the 40th IEEE Conference on Decision and Control, 2001.*, pages 205 –210 vol.1.

[17] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6), june 2006.

[18] M. Braverman. Towards deterministic tree code constructions. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012.

[19] M. Braverman and A. Rao. Towards coding for maximum errors in interactive communication. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, 2011.

[20] A. Budhiraja and H. J. Kushner. Monte carlo algorithms and asymptotic problems in nonlinear filtering. In *Stochastics in Finite/Infinite Dimensions (Volume in honor of Gopinath Kallianpur)*, 1999.

[21] O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag, New York, Inc., 2005.

[22] D. Crisan and A. Doucet. Convergence of sequential monte carlo methods. Technical report, Sequential Monte Carlo Methods in Practice, 2000.

[23] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, 2002.

[24] R. Curry. *Estimation and Control with Quantized Measurements*. M.I.T Press, 1970.

[25] R. Curry, W.V Velde, and J. Potter. Nonlinear estimation with quantized measurements - pcm predictive quantization, and data compression. *Information Theory, IEEE Transactions on*, 16(2):152–161, 1970.

[26] J. C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State-space solutions to standard h2 and h-infinity control problems. *IEEE Transactions on Automatic Control*, 34:831 –847, aug 1989.

[27] W. Evans, M. Klugerman, and L.J. Schulman. Constructive tree codes with polynomial size alphabet. (unpublished manuscript).

[28] R. Fano. A heuristic discussion of probabilistic decoding. *IEEE Transactions on Information Theory*, 1963.

[29] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright. Lp decoding corrects a constant fraction of errors. *IEEE Transactions on Information Theory*, 2007.

[30] J. Feldman, M.J. Wainwright, and D.R. Karger. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 2005.

[31] G. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Pearson Prentice Hall, 5th, edition, 2006.

[32] R. Gallager. Low-density parity-check codes. *Information Theory, IRE Transactions on*, 1962.

[33] R. G. Gallager. *Information Theory and Reliable Communication*. 1968.

[34] R. Gelles and A. Sahai. Potent tree codes and their applications: Coding for interactive communication, revisited. *CoRR*, abs/1104.0739, 2011.

[35] M. G. Genton. *Skew-Elliptical Distributions and their Applications —A Journey beyond Normality*. Chapman and Hall/CRC, 2004.

[36] M. G. Genton. Discussion of the skew normal. *Scandinavian Journal of Statistics*, 32(2):189–198, 2005.

[37] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings-F*, 140(2):107–113, 1993.

[38] O. Güler and F. Gürtuna. The extremal volume ellipsoids of convex bodies, their symmetry properties, and their determination in some special cases. *arXiv*, math.MG, Sep 2007.

[39] V. Guruswami. Reed-solomon and algebraic-geometric codes. In *List Decoding of Error-Correcting Codes*, volume 3282 of *Lecture Notes in Computer Science*, pages 219–236. Springer Berlin / Heidelberg, 2005.

[40] B. Hassibi, A.H. Sayed, and T. Kailath. *Indefinite-Quadratic Estimation and Control*. SIAM, 1999.

[41] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Transactions on Automatic Control*, 50(11), Nov. 2005.

[42] J.K. Hedrick, M. Tomizuka, and P. Varaiya. Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14, 1994.

[43] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: a programmable pervasive space. *Computer*, 38, march 2005.

[44] J.P. Hespanha, P. Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138 –162, Jan. 2007.

[45] O. C. Imer, S. Yksel, and T. Baar. Optimal control of lti systems over unreliable communication links. *Automatica*, 42(9), 2006.

[46] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6), june 2003.

[47] Wozencraft J.M.R. *Sequential decoding for reliable communication*. Research Laboratory of Electronics, MIT, 1957.

[48] Wozencraft J.M.R. and Reiffen B. *Sequential Decoding*. The Technology Press of MIT, 1961.

[49] T. Kailath. *Linear Systems*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980.

[50] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 2000.

[51] A. M. Kakhaki, H. K. Abadi, P. Pad, H. Saeedi, K. Alishahi, and F. Marvasti. Capacity achieving random sparse linear codes. *CoRR*, abs/1102.4099, 2011.

[52] R. E. Kalman. On the general theory of control systems. *in Proc. of the first IFAC Congress, London*, 1:481–491, 1960.

[53] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95–108, 1961.

[54] G. R. Karlsson and F. Gustafsson. Particle filtering for quantized sensor information. In *13th European Signal Processing Conference, EUSIPCO*. EURASIP, Turkey, 2005.

[55] A. Krasnopeev, J. Xiao, and Z. Luo. Minimum energy decentralized estimation in sensor network with correlated sensor noise. In *ICASSP*, volume 3, pages iii/673–iii/676, 2005.

[56] P. R. Kumar. Experimental results involving delay. `https://netfiles.uiuc.edu/prkumar/www/testbed/videos/700msU2KD1.mpg`.

[57] E.A. Lee. Cyber physical systems: Design challenges. *IEEE International Symposium on Object Oriented Real-Time Distributed Computing*, 2008.

[58] A. Lévy. *Basic Set Theory*. Number v. 13. Dover Publications, 2002.

[59] Z. Luo and J. Xiao. Universal decentralized estimation in a bandwidth constrained sensor network. In *ICASSP*, volume 4, pages iv/829–iv/832, 2005.

[60] L. He M. G. Genton and X. Liu. Moments of skew-normal random vectors and their quadratic forms. *Statistics and Probability Letters*, 51(4):319–325, 2001.

[61] D.J. C. MacKay and R.M. Neal. Near shannon limit performance of low density parity check codes. *Electronics Letters*, 32, Aug 1996.

[62] N.C. Martins, M.A. Dahleh, and N. Elia. Feedback stabilization of uncertain systems in the presence of a direct link. *IEEE Transactions on Automatic Control*, 51(3), march 2006.

[63] J. Massey. Shift-register synthesis and bch decoding. *IEEE Transactions on Information Theory*, 15, Jan 1969.

[64] Alexey S. Matveev and Andrey V. Savkin. *Estimation and Control over Communication Networks (Control Engineering)*. Birkhauser, 2007.

[65] M. Mesbahi and F.Y. Hadaegh. Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching. *Journal of Guidance, Control and Dynamics*, 24, 2001.

[66] P. Minero, M. Franceschetti, S. Dey, and G.N. Nair. Data rate theorem for stabilization over time-varying feedback channels. *IEEE Transactions on Automatic Control*, 54(2), 2009.

[67] Y. Mo and B. Sinopoli. A characterization of the critical value for kalman filtering with intermittent observations. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, dec. 2008.

[68] A. Moitra. Efficiently coding for interactive communication. *Electronic Colloquium on Computational Complexity (ECCC)*, 2011.

[69] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2), feb. 2005.

[70] R.M. Murray, K.J. Astrom, S.P. Boyd, R.W. Brockett, and G. Stein. Future directions in control in an information-rich world. *Control Systems, IEEE*, 23(2), apr 2003.

[71] G.N. Nair and R.J. Evans. Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM Journal on Control and Optimization*, 43(2):413–436, 2005.

[72] H. Nyquist. Regeneration theory. *Bell System Technical Journal*, 11, 1932.

[73] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), jan. 2007.

[74] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), Sept. 2004.

[75] M. G. P. Naveau, Genton and X. Shen. A skewed kalman filter. *J. Multivar. Anal.*, 94(2):382–400, 2005.

[76] H. Palaiyanur and A. Sahai. A simple encoding and decoding strategy for stabilization over discrete memoryless channels. *Proceedings of the Allerton Conference on Control, Communication and Computing*, 2005.

[77] C. H. Papadimitriou and J. N. Tsitsiklis. Intractable problems in control theory. In *Decision and Control, 1985 24th IEEE Conference on*, volume 24, dec. 1985.

[78] S. Y. Park and A. Sahai. Intermittent kalman filtering: Eigenvalue cycles and nonuniform sampling. In *American Control Conference (ACC), 2011*, 29 2011-july 1 2011.

[79] S. Rajagopalan and L. Schulman. A coding theorem for distributed computation. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, STOC '94. ACM, 1994.

[80] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis. Soi-kf: Distributed kalman filtering with low-cost communications using the sign of innovations. *IEEE Transactions on Signal Processing*, 54(12):4782–4795, 2006.

[81] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47, feb 2001.

[82] T.J. Richardson and R.L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47, feb 2001.

[83] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51, feb. 2006.

[84] A. Sahai. *Anytime Information Theory*. PhD thesis, Massachusetts Institute of Technology, 2001.

[85] A. Sahai. Why do block length and delay behave differently if feedback is present? *IEEE Transactions on Information Theory*, 54(5), May 2008.

[86] A. Sahai and S. Mitter. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link - part i: Scalar systems. *IEEE Transactions on Information Theory*, 52(8):3369–3395, 2006.

[87] Anant Sahai and Sanjoy K. Mitter. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link, part ii: vector systems. *http://arxiv.org/abs/cs/0610146*, abs/cs/0610146, 2006.

[88] N. Sandell Jr, P. Varaiya, M. Athans, and M. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, 23, apr 1978.

[89] Igal Sason and Shlomo Shamai. Performance analysis of linear codes under maximum-likelihood decoding: A tutorial. *FNT in Communications and Information Theory*, 3(1/2):1–222, 2006.

[90] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S.S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1), jan. 2007.

[91] L.J. Schulman. `http://users.cms.caltech.edu/~schulman/Papers/intercodingpostscript.txt%`.

[92] L.J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745 – 1756, 1996.

[93] F. Schweppe. Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13(1), February 1968.

[94] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379 – 423 and 623 – 656, July and Oct 1948.

[95] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. In *Proceedings. 42nd IEEE Conference on Decision and Control, 2003*, volume 1, pages 701–708 Vol.1, 2003.

[96] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9), Sept. 2004.

[97] A. Sluis. Upperbounds for roots of polynomials. *Numerische Mathematik*, 15(3):250–262, 1970.

[98] R.T. Sukhavasi and B. Hassibi. The kalman-like particle filter: Optimal estimation with quantized innovations/measurements. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 4446 –4451, 15-18 2009.

[99] R.T. Sukhavasi and B. Hassibi. Particle filtering for quantized innovations. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.*, pages 2229 –2232, 19-24 2009.

[100] S. Tatikonda and S. Mitter. Control under communication constraints. *IEEE Transactions on Automatic Control*, 49(7):1056– 1068, 2004.

[101] S. Tatikonda, A. Sahai, and S. Mitter. Stochastic linear control over a communication channel. *IEEE Transactions on Automatic Control*, 49(9):1549 – 1561, Sept. 2004.

[102] J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9), sep 1986.

[103] J. N Tsitsiklis. Problems in decentralized decision making and computation. 1984.

[104] R. van Handel. Uniform time average consistency of monte carlo particle filters. *Stochastic Processes and their Applications*, 119(11), 2009.

[105] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13, April 1967.

[106] P. O. Vontobel and Ralf Koetter. Graph-cover decoding and finite-length analysis of message-passing iterative decoding of ldpc codes. *CoRR*, abs/cs/0512078, 2005.

[107] Ren W. and R.W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5), May 2005.

[108] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Technology Press and Wiley, NY, 1949.

[109] H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6, 1968.

[110] W. S. Wong and R.W. Brockett. Systems with finite communication bandwidth constraints. i. state estimation problems. *IEEE Transactions on Automatic Control*, 42(9):1294–1299, 1997.

[111] W. S. Wong and R.W. Brockett. Systems with finite communication bandwidth constraints. ii. stabilization with limited information feedback. *IEEE Transactions on Automatic Control*, 44, May 1999.

[112] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1), 2004.

[113] K. You, L. Xie, S. Sun, and W. Xiao. Multiple-level quantized innovation kalman filter. In *17th International Federation of Automatic Control*. Seoul, Korea, 2008.

[114] S. Yüksel. A random time stochastic drift result and application to stochastic stabilization over noisy channels. *47th Annual Allerton Conference on Communication, Control, and Computing, 2009. Allerton 2009.*, 2009.

[115] S. Yüksel. A random time stochastic drift result and application to stochastic stabilization over noisy channels. *47th Annual Allerton Conference on Communication, Control, and Computing, 2009. Allerton 2009.*, pages 628–635, 2009.

[116] S. Yüksel. Stochastic stability of adaptive quantizers for markov sources. *IEEE International Symposium on Information Theory, 2009.*, pages 527–531, 2009.

[117] S. Yüksel. Existence of optimal average cost optimal lqg coding and control policies over discrete channels: Stable and unstable cases. 2010.

[118] S. Yüksel and T. Başar. Control over noisy forward and reverse channels. *IEEE Transactions on Automatic Control*, 56(5), May 2011.