Efficient Communication over Additive White Gaussian Noise and Intersymbol Interference Channels Using Chaotic Sequences

Brian Chen

RLE Technical Report No. 598

April 1996

The Research Laboratory of Electronics
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139-4307

This work was supported in part by the Department of the Navy, Office of the Chief of Naval Research, under Contract N0014-93-1-0686 as part of the Advanced Research Projects Agency's RASSP program and under Grant N00014-95-1-0834, and in part by a National Defense Science and Engineering Graduate Fellowship.

Efficient Communication over Additive White Gaussian Noise and Intersymbol Interference Channels Using Chaotic Sequences

by

Brian Chen

Submitted to the Department of Electrical Engineering and Computer Science
on January 19, 1996, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Abstract

Chaotic signals and systems are potentially attractive in a variety of signal modeling and signal synthesis applications. State estimation algorithms for chaotic sequences corresponding to tent map dynamics in the presence of additive Gaussian noise, intersymbol interference, and multiple access interference are developed and their performance is evaluated both analytically and empirically. These algorithms may be useful in a wide range of noise removal, deconvolution, and signal separation contexts. In the additive Gaussian noise case, the previously derived maximum likelihood estimator in stationary white noise is extended to the case of nonstationary white noise. Useful analytic expressions for performance evaluation of both estimators are derived, and the estimators are shown to be asymptotically unbiased at high SNR and to have an error variance which decays exponentially with sequence length to a lower threshold which depends on the SNR. An estimator for the colored noise case is also suggested. In the intersymbol interference case, three new iterative algorithms are developed and their performance is evaluated empirically, considering the effects of sequence length, noise, and initialization. Finally, in the multiple access interference case, an optimal maximum likelihood estimator for the separation of superimposed tent map sequences is derived, along with the associated Cramer-Rao bound on the error variance. The estimator is asymptotically unbiased at high SNR and has an error variance which depends on the particular tent map sequences.

As an example of a potential application of chaotic signals in communication, a new paradigm for exploiting chaos in the modulation or coding of analog sources is explored in which the source data is encoded in the initial value of a tent map sequence. It is shown that this so-called tent map coding system can be interpreted as a twisted modulation system, and its performance is analyzed in this context. Also, this coding system is shown to outperform linear modulation codes and M-ary digital codes in a range of power-bandwidth regimes.

Thesis Supervisor: Gregory W. Wornell
Title: Assistant Professor of Electrical Engineering

Acknowledgments

First, of course, I would like to thank my advisor, Prof. Greg Wornell, for his patience, guidance, helpful insights, and useful advice during both the research and writing phases. Greg, thanks for taking the time to read the *entire* manuscript — over your holiday, no less! I hope it wasn't too painful.

I also gratefully acknowledge the Office of Naval Research and the National Defense Science and Engineering Graduate Fellowship Program for their financial contributions.

I would also like to recognize my instructors at MIT. Much of the background for the work in this thesis was taught to me by the professors and TAs in my courses here. I thank all of you for your dedication to teaching.

I would like to acknowledge my fellow graduate students at MIT, especially the students of the Digital Signal Processing Group, for the many perspectives and insights gained through informal conversations. Let me also thank the members of "thepack@mit" for the welcome social relief from the day-to-day rigors of graduate student life.

I would also like to acknowledge the professors at my undergraduate institution, the University of Michigan. You helped me get started in electrical engineering and taught me the fundamentals. Go Blue!

Finally, and most importantly, I would like to thank my family for all of their help and support. Nancy, thanks for leading the way and showing me the ropes. Mom, thanks for all the loving care that only a mother can provide. Your Thanksgiving and Christmas dinners kept me going all year long. Dad, you're not only a gentleman and a scholar, but also my role model, confidant, and most trusted advisor — a true hero.

To any brave souls

who actually read this thesis from start to finish.

I salute you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chaotic systems are potentially attractive in a variety of signal modeling and signal synthesis applications. In many of these applications, the chaotic signals of interest need to be recovered from corrupted signals which have undergone various forms of distortion. Thus, the work in this thesis involves developing estimation algorithms for chaotic sequences in the presence of additive Gaussian noise, intersymbol interference (ISI), and multiple access interference. In certain contexts, these algorithms may also be referred to as noise removal algorithms, deconvolution algorithms, and signal separation algorithms, respectively.

Although these algorithms may have many different applications, this thesis explores more thoroughly the application of chaotic signals and systems for communication. As an example of such an application, this thesis examines the potential role of discrete-time chaotic systems in coding and modulation and in deconvolution. In each of these applications, the underlying deterministic structure of the sequences generated by chaotic systems is exploited to achieve noise or ISI reduction.

The systems of interest in this thesis are one-dimensional dynamic systems whose state evolutions correspond to iterated maps. In particular, they have a single state variable $x[n]$ whose behavior is governed by a mapping from the previous state to the current state, i.e.,

$$x[n] = F(x[n-1]), \tag{1.1}$$

where $F(\cdot)$ represents a chaotic mapping with sensitivity to initial condition, i.e., small perturbations in the initial state $x[0]$ lead to divergent state trajectories $x[n]$ [7]. While, in general, the output of a dynamic system may be a complicated function of its state, in all of the applications in this thesis, the output of the chaotic system is simply the state variable $x[n]$ itself.

Much of the thesis focuses on the class of chaotic systems whose state evolution corresponds to the so-called symmetric tent map. The chaotic mapping in these systems is

$$F(x) = \beta - 1 - \beta|x|, \qquad 1 < \beta \leq 2. \tag{1.2}$$

The piecewise linearity of this map makes this class of chaotic systems particularly amenable to analysis. The sequences $x[n]$ generated by such systems are referred to as tent map sequences, and tent map sequences in the $\beta = 2$ case have a natural interpretation as quantizers of their initial states. More precisely, if the initial state $x[0]$ is in the interval $[-1, 1]$, the sign of $x[n]$ represents the $(n+1)$th most significant bit in the quantization of $x[0]$, and the real number $x[n]$ represents the position of $x[0]$ within the particular quantization region. This thesis often specializes results to the special case $\beta = 2$.

Throughout the thesis it is necessary to view chaotic sequences as deterministic in some situations and random in others. Clearly, given an initial condition $x[0]$, any sequence of the form (1.1) is completely determined. However, due to the sensitivity to initial condition characteristic mentioned above, iteration of the map (1.2) using finite-precision calculations quickly leads to unpredictable results. Thus, it is sometimes useful to treat these sequences as sequences of random variables. One can associate with these sequences an invariant density, the probability density such that if the initial condition is randomly drawn from this density, the subsequent $x[n]$ will all also have this density. For example, tent map sequences in the case $\beta = 2$ have an invariant density which is uniform on the interval $[-1, 1]$. Furthermore, for almost all initial conditions in this interval, the sequences are ergodic and uncorrelated [5].

The first part of the thesis examines the application of chaotic sequences to communication over the additive white Gaussian noise (AWGN) channel. Chapter 2 extends the maximum likelihood (ML) estimator for tent map sequences in stationary AWGN [5] to the case of nonstationary AWGN and analyzes the performance of the estimators in both cases analytically and empirically. Chapter 3 explores a paradigm for exploiting tent map sequences in the modulation or coding of analog sources for transmission over the AWGN channel and analyzes the performance of this coding system relative to linear modulation codes and M-ary digital codes.

The second part of the thesis examines the application of chaotic sequences to communication over nonideal channels, i.e., channels such as ISI channels and multiple access channels which corrupt the input signals with more than simple AWGN. Chapter 4 deals with deconvolution or equalization in which the underlying deterministic structure of chaotic sequences is exploited to reduce ISI. This chapter develops three deconvolution algorithms, analyzes their performance, and examines these algorithms in the context of decoding for the ISI channel. Finally, Chapter 5 develops the ML estimator for tent map sequences that have been transmitted over a multiple access channel. This algorithm is motivated in a multiple access communication context, but may be more generally applicable in other signal separation problems as well.

Chapter 2

Estimation of Tent Map Sequences

To fully exploit chaotic sequences for the communications applications in this thesis, algorithms for the estimation of these sequences in the presence of noise are needed. Papadopoulos and Wornell have developed the ML estimator for tent map sequences in the the presence of stationary AWGN [5]. This chapter extends their work to the more general case of nonstationary AWGN and suggests an approach which may also be useful for the general colored Gaussian noise case. An analysis of the relationship between signal to noise ratio (SNR) and the mean square estimation error in the case of stationary AWGN is also presented.

2.1 The ML Estimator in White Gaussian Noise

This section considers the estimation of tent map sequences in the following scenario. A sequence of length N, denoted by

$$x[0], x[1], \cdots, x[N-1] \tag{2.1}$$

is generated via (1.1) with (1.2). The estimator observes a noise-corrupted version of this sequence, denoted $y[0], y[1], \cdots, y[N-1]$ where

$$y[k] = x[k] + w[k] \tag{2.2}$$

and $w[k]$ is a sequence of jointly Gaussian random variables with a diagonal covariance matrix K_{ww}, i.e.,

$$K_{ww} = \mathrm{diag}(\sigma^2_{w[0]}, \sigma^2_{w[1]}, \cdots, \sigma^2_{w[N-1]}). \tag{2.3}$$

Based on observation of the $y[k]$, the estimator produces the optimal estimate of $x[n]$ under the maximum likelihood criteria.

Such an estimator for the special case where $K_{ww} = \sigma_w^2 I$ is derived in [5]. In the more general case where K_{ww} is some arbitrary diagonal covariance matrix, the estimator has a similar form. Specifically, the estimator can be decomposed into a recursive filtering stage followed by a smoothing stage. The next section derives the ML estimator for this more general case.

2.2 ML Estimator Derivation

The derivation of the ML estimatior involves three main components:

1. Proof that filtering and smoothing can be separated.
2. Derivation of filtering stage.
3. Derivation of smoothing stage.

The following function and its inverse used in [5] aid in the general derivation:

$$F_s(x) = \beta - 1 - \beta s x \tag{2.4}$$

and

$$F_s^{(-1)}(x) = \frac{\beta - 1 - x}{\beta} \cdot s. \tag{2.5}$$

Also, $F^{(k)}_{s_1,\cdots,s_{k-1},s_k}(\cdot)$ and $F^{(-k)}_{s_1,s_2,\cdots,s_k}(\cdot)$ denote the k-fold iteration of (2.4) and (2.5), respectively, i.e.,

$$F^{(k)}_{s_1,\cdots,s_{k-1},s_k}(x) = F_{s_k} \circ F_{s_{k-1}} \circ \cdots \circ F_{s_1}(x) \tag{2.6}$$

and

$$F^{(-k)}_{s_1,s_2,\cdots,s_k}(x) = F^{(-1)}_{s_1} \circ F^{(-1)}_{s_2} \circ \cdots \circ F^{(-1)}_{s_k}(x). \tag{2.7}$$

These definitions imply

$$F(x) = F_{\mathrm{sgn}\ x}(x) \tag{2.8}$$

and

$$\begin{aligned} x[n] &= F^{(-m)}_{s[n],s[n+1],\cdots,s[n+m-1]}(x[n+m]) \\ &= F^{(m)}_{s[n-m],s[n-m+1],\cdots,s[n-1]}(x[n-m]), \end{aligned} \tag{2.9}$$

where $s[n] = \mathrm{sgn}\ x[n]$ and $m > 0$. In particular, letting $m = N - 1 - n$, one finds that

$$s[0], s[1], \cdots, s[N-2], x[N-1] \tag{2.10}$$

is an equivalent representation for (2.1) obtained through the nonlinear transformation (2.9).

2.2.1 Separation of Filtering and Smoothing

This section demonstrates that the ML estimator in AWGN for sequences whose dynamics correspond to the so-called tent map can be decomposed into a filtering stage followed by a smoothing stage.

ML estimation is invariant to invertible nonlinear transformations, so

$$\hat{x}[n|N-1] = F^{(n-(N-1))}_{\hat{s}[n|N-1],\hat{s}[n+1|N-1],\cdots,\hat{s}[N-2|N-1]}(\hat{x}[N-1|N-1]), \tag{2.11}$$

where $\hat{x}[n|k]$ denotes the best estimate of $x[n]$ based on observation of $y[0], \cdots, y[k]$. Also, for any $m > 0$, $F^{(n+m)}(x) = F^{(n+m)}(-x)$, so the probability density of $y[n+m]$ does not depend on $s[n]$. Thus,

$$\begin{aligned} \hat{s}[n|N-1] &= \hat{s}[n|y[0], y[1], \cdots, y[N-1]] \\ &= \hat{s}[n|y[0], y[1], \cdots, y[n]] \\ &= \hat{s}[n|n] \\ &= \widehat{\mathrm{sgn}}\ x[n|n] \end{aligned}$$

$$= \quad \text{sgn}\ \hat{x}[n|n] \equiv \hat{s}[n], \tag{2.12}$$

i.e., the estimates $\hat{s}[n]$ can be obtained causally. Therefore, the ML estimator can be decomposed into two stages. The first stage finds the filtered estimates $\hat{x}[n|n]$ for $0 \le n \le N-1$, and the second stage produces the smoothed estimates $\hat{x}[n|N-1]$ through (2.12) and (2.11). Specifically,

$$\begin{aligned}\hat{x}[n|N-1] &= F^{(n-(N-1))}_{\hat{s}[n|N-1],\hat{s}[n+1|N-1],\cdots,\hat{s}[N-2|N-1]}(\hat{x}[N-1|N-1]) \\ &= F^{(-1)}_{\text{sgn}(\hat{x}[n|n])}(\hat{x}[n+1|N-1]). \end{aligned} \tag{2.13}$$

2.2.2 Filtering

The ML filtering criterion involves jointly maximizing the likelihood function of the observations

$$\underset{s[0],\cdots,s[n-1],x[n]}{\arg\max} \quad \ln p(y[0],\cdots,y[n]|s[0],\cdots,s[n-1],x[n]) \tag{2.14}$$

over the arguments $s[0], s[1], \cdots, s[n-1], x[n]$. Since $y[n]$ is independent of $s[k]$ for $k < n$, by induction one can perform the maximization (2.14) over the $s[k]$ separately from the maximization over $x[n]$. The induction proceeds as follows.

1. If one is given the $\hat{s}[0], \cdots, \hat{s}[n-1]$ which maximize the likelihood of the observations $y[0], \cdots, y[n-1]$, one can maximize the likelihood of the observations $y[0], \cdots, y[n]$ over only $x[n]$.

   To see this, first introduce the following vectors for notational convenience:

$$\mathbf{s}_{n-1} = [s[0] \ \cdots \ s[n-1]]^T \tag{2.15}$$

$$\mathbf{y}_n = [y[0] \ \cdots \ y[n]]^T. \tag{2.16}$$

   Then,

$$\ln p(\mathbf{y}_n|\mathbf{s}_{n-1}, x[n]) = \sum_{k=0}^{n} \ln p(y[k]|\mathbf{s}_{n-1}, x[n])$$

$$\begin{aligned}
&= \ln p(y[n]|x[n]) + \sum_{k=0}^{n-1} \ln p(y[k]|\mathbf{s}_{n-1}, x[n]) \\
&= \ln p(y[n]|x[n]) + \ln p(\mathbf{y}_{n-1}|\mathbf{s}_{n-1}, x[n]) \qquad (2.17)
\end{aligned}$$

and, thus,

$$\arg\max_{\mathbf{s}_{n-1}, x[n]} \ln p(\mathbf{y}_n|\mathbf{s}_{n-1}, x[n]) = \{\hat{\mathbf{s}}_{n-1}, \arg\max_{x[n]} \ln p(\mathbf{y}_n|\hat{\mathbf{s}}_{n-1}, x[n])\}. \qquad (2.18)$$

2. Initialize the induction with

$$\hat{s}[0] = \operatorname{sgn} \hat{x}[0|0] = \operatorname{sgn} y[0]. \qquad (2.19)$$

Thus, the filtering problem reduces to maximizing the likelihood function over the single unknown $x[n]$. The following identity is useful in performing the maximization:

$$a - F_s^{(-1)}(b) = -s\frac{F_s(a) - b}{\beta}, \qquad s = \pm 1. \qquad (2.20)$$

By repeatedly applying this identity, one obtains

$$\left(y[i] - F^{(i-n)}_{\hat{s}[i],\cdots,\hat{s}[n-1]}(x[n])\right)^2 = \left(\frac{1}{\beta}\right)^{2(n-i)} \left(F^{(n-i)}_{\hat{s}[i],\cdots,\hat{s}[n-1]}(y[i]) - x[n]\right)^2. \qquad (2.21)$$

Then,

$$\begin{aligned}
\hat{x}[n|n] &= \arg\max_{x[n]} \ln p(\mathbf{y}_n|\hat{\mathbf{s}}_{n-1}, x[n]) \\
&= \arg\min_{x[n]} \sum_{i=0}^{n} \left(\frac{y[i] - F^{(i-n)}_{\hat{s}[i],\cdots,\hat{s}[n-1]}(x[n])}{\sigma_w[i]}\right)^2 \\
&= \arg\min_{x[n]} \sum_{i=0}^{n} \left(\frac{1}{\beta^{2(n-i)}\sigma_w^2[i]}\right) \left(F^{(n-i)}_{\hat{s}[i],\cdots,\hat{s}[n-1]}(y[i]) - x[n]\right)^2. \qquad (2.22)
\end{aligned}$$

The minimization (2.22) is equivalent to minimizing a suitably weighted norm of the error $\mathbf{e}$ in the following system of equations:

$$\underbrace{\begin{bmatrix} F^{(n)}_{\hat{s}[0],\cdots,\hat{s}[n-1]}(y[0]) \\ F^{(n-1)}_{\hat{s}[1],\cdots,\hat{s}[n-1]}(y[1]) \\ \vdots \\ F^{(0)}(y[n]) \end{bmatrix}}_{\mathbf{y}'} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{\mathbf{1}} x[n] + \mathbf{e}. \tag{2.23}$$

In particular, the norm to be minimized is $||\mathbf{e}||^2 = \mathbf{e}^T \mathbf{S}_n \mathbf{e}$, where

$$\mathbf{S}_n = \text{diag}(s_{0,n}, s_{1,n}, \cdots, s_{n,n}) \tag{2.24}$$

and where

$$s_{i,n} = \frac{1}{\beta^{2(n-i)}\sigma^2_{w[i]}}. \tag{2.25}$$

Then, the ML filtered estimate is [3]

$$\begin{aligned} \hat{x}[n|n] &= \arg\min_{x[n]} ||\mathbf{y}' - \mathbf{1}x[n]||^2 \\ &= (\mathbf{1}^T\mathbf{S}_n\mathbf{1})^{-1}\mathbf{1}^T\mathbf{S}_n\mathbf{y}' \\ &= \frac{\sum_{i=0}^n \frac{1}{\sigma^2_{w[i]}\beta^{2(n-i)}} F^{(n-i)}_{\hat{s}[i],\hat{s}[i+1],\cdots,\hat{s}[n-1]}(y[i])}{\sum_{i=0}^n \frac{1}{\sigma^2_{w[i]}\beta^{2(n-i)}}}. \end{aligned} \tag{2.26}$$

A convenient recursion for the filtered estimates exists and is derived below. Defining the following quantities aids in the derivation. Let

$$a_{i,n} \equiv \frac{s_{i,n}}{\sum_{j=0}^n s_{j,n}}. \tag{2.27}$$

Note that for any n, $\sum_{i=0}^{n-1} a_{i,n-1} = 1$ so that [5]

$$\sum_{i=0}^{n-1} a_{i,n-1} F_s(u_i) = F_s\left(\sum_{i=0}^{n-1} a_{i,n-1} u_i\right). \tag{2.28}$$

Next, define

$$\begin{aligned}
\gamma &\equiv \frac{a_{i,n}}{a_{i,n-1}} && (2.29)\\
&= \frac{s_{i,n}}{s_{i,n-1}} \frac{\sum_{j=0}^{n-1} s_{j,n-1}}{\sum_{j=0}^{n} s_{j,n}} \\
&= \frac{\beta^{-2} \sum_{j=0}^{n-1} s_{j,n-1}}{\sum_{j=0}^{n-1} s_{j,n} + s_{n,n}} \\
&= \frac{\sum_{j=0}^{n-1} s_{j,n}}{\sum_{j=0}^{n-1} s_{j,n} + s_{n,n}}, && (2.30)
\end{aligned}$$

where the third equality arises from the fact that $\frac{s_{i,n}}{s_{i,n-1}} = \beta^{-2}$ for $i < n$. Then, finally, the recursion for the filtered estimates $\hat{x}[n|n]$ is

$$\begin{aligned}
\hat{x}[n|n] &= \sum_{i=0}^{n} a_{i,n} F^{(n-i)}_{\hat{s}[i],\hat{s}[i+1],\cdots,\hat{s}[n-1]}(y[i]) \\
&= \gamma \sum_{i=0}^{n-1} a_{i,n-1} F_{\hat{s}[n-1]} \left(F^{(n-1-i)}_{\hat{s}[i],\cdots,\hat{s}[n-2]}(y[i]) \right) + a_{n,n} y[n] \\
&= \gamma F_{\hat{s}[n-1]} \left(\sum_{i=0}^{n-1} a_{i,n-1} F^{(n-1-i)}_{\hat{s}[i],\cdots,\hat{s}[n-2]}(y[i]) \right) + a_{n,n} y[n] \\
&= \gamma F_{\hat{s}[n-1]}(\hat{x}[n-1|n-1]) + a_{n,n} y[n] \\
&= \gamma F(\hat{x}[n-1|n-1]) + a_{n,n} y[n] \\
&= \frac{\sum_{i=0}^{n-1} s_{i,n}}{\sum_{i=0}^{n-1} s_{i,n} + s_{n,n}} F(\hat{x}[n-1|n-1]) + \frac{s_{n,n}}{\sum_{i=0}^{n-1} s_{i,n} + s_{n,n}} y[n], && (2.31)
\end{aligned}$$

with the initialization $\hat{x}[0|0] = y[0]$. The second line follows from the definition of γ (2.29), the third from (2.28), and the final line from (2.30). As a final remark, note that the filtered estimate in (2.31) does not use the a priori knowledge that $x[n] \in [-1, \beta - 1]$ for all n as long as $x[0] \in [-1, \beta - 1]$. Therefore, one should amplitude limit the estimates $\hat{x}[n|n]$ given by (2.31) to this interval.

2.2.3 Smoothing

Recall, the smoothed estimates $\hat{x}[n|N-1]$ are given in Subsection 2.2.1 by (2.13). For convenience, this expression is repeated here:

$$\begin{aligned}
\hat{x}[n|N-1] &= F^{(n-(N-1))}_{\hat{s}[n|N-1],\hat{s}[n+1|N-1],\cdots,\hat{s}[N-2|N-1]}(\hat{x}[N-1|N-1]) \\
&= F^{(-1)}_{\mathrm{sgn}(\hat{x}[n|n])}(\hat{x}[n+1|N-1]). \qquad (2.32)
\end{aligned}$$

2.3 Error Analysis

This section analyzes the filtering and smoothing errors for the estimates given by (2.31) and (2.13). Useful approximate expressions for the mean square estimation error are derived. In the case where the noise is stationary, these expressions are compared to associated Cramer-Rao bounds and experimental results.

2.3.1 Filtering Error

One can compute good approximations to the error variance of the ML filtered estimates directly from (2.31). Denote the filtering errors by $e[n|n] = \hat{x}[n|n] - x[n]$ and denote their variances by $\sigma^2[n|n]$.

Bias

If the SNR is high enough such that $\hat{s}[n] = s[n]$ with high probability, one can show by induction that the ML filtered estimates (2.31) are approximately unbiased.

1. Assume that $e[n-1|n-1]$ has zero mean. Then,

$$\begin{aligned}
E\{\hat{x}[n|n]\} &= E\{(1-a_{n,n})F(\hat{x}[n-1|n-1]) + a_{n,n}y[n]\} \\
&= E\Big\{(1-a_{n,n})F_{\hat{s}[n-1]}(x[n-1]+e[n-1|n-1]) \\
&\quad + a_{n,n}(x[n]+w[n])\Big\} \\
&= E\{(1-a_{n,n})(\beta - 1 - \beta\hat{s}[n-1]x[n-1] \\
&\quad - \beta\hat{s}[n-1]e[n-1|n-1])\} + a_{n,n}x[n]
\end{aligned}$$

$$\begin{aligned}
&\approx (1-a_{n,n})F_{s[n-1]}(x[n-1]) + a_{n,n}x[n] \\
&\quad - (1-a_{n,n})\beta s[n-1]E\{e[n-1|n-1]\} \\
&= (1-a_{n,n})x[n] + a_{n,n}x[n] \\
&= x[n]. \qquad (2.33)
\end{aligned}$$

The approximation in the fourth line arises from substituting $s[n-1]$ for $\hat{s}[n-1]$. Thus, if $e[n-1|n-1]$ has zero mean at high SNR, so does $e[n|n]$.

2. To complete the induction, note that

$$E\{e[0|0]\} = E\{w[0]\} = 0. \qquad (2.34)$$

As a final note, the above claim is supported by experimental results in [5] for the case of stationary AWGN.

Variance

One can compute the approximate error variance by making the same assumptions as above, namely that the SNR is high enough such that $\hat{s}[n] \approx s[n]$.

$$\begin{aligned}
\sigma^2[n|n] &= E\{(\hat{x}[n|n] - x[n])^2\} \\
&= E\left\{[(1-a_{n,n})(F(\hat{x}[n-1|n-1]) - x[n]) + a_{n,n}(y[n]-x[n])]^2\right\} \\
&\approx E\left\{[(1-a_{n,n})(-\beta s[n-1]e[n-1|n-1]) + a_{n,n}w[n]]^2\right\} \\
&= (1-a_{n,n})^2\beta^2\sigma^2[n-1|n-1] + a_{n,n}^2\sigma_{w[n]}^2. \qquad (2.35)
\end{aligned}$$

Thus, one can recursively compute the error variances with the initialization $\sigma^2[0|0] = \sigma_{w[0]}^2$. In the case of stationary noise, where $\sigma_{w[n]}^2 = \sigma_w^2$ for all n,

$$\sigma^2[n|n] = a_{n,n}\sigma_w^2 = \frac{1-\beta^{-2}}{1-\beta^{-2(n+1)}}\sigma_w^2. \qquad (2.36)$$

This approximate expression, which can be verified by direct substitution into (2.35), equals the Cramer-Rao bound derived in [5] for the case of stationary AWGN. Also in [5], this bound is shown experimentally to be tight at high SNR. One should note that the Cramer-Rao bound is not a true lower bound at low SNR, since it neglects the fact that the sequence $x[n]$ is known a priori to lie in the interval $[-1, \beta - 1]$. In particular, the error variance does not grow to infinity in the limit of large σ_w^2 as suggested by (2.36).

2.3.2 Smoothing Error

In view of (2.13) and (2.5), one might try to approximate the error variance of the smoothed estimates in the following way:

$$\sigma^2[n|N-1] \approx \frac{\sigma^2[n+1|N-1]}{\beta^2} \approx \frac{\sigma^2[N-1|N-1]}{\beta^{2(N-1-n)}}. \tag{2.37}$$

In the case of stationary AWGN, this reduces to

$$\sigma^2[n|N-1] \approx \frac{1-\beta^{-2}}{1-\beta^{-2N}} \sigma_w^2 \beta^{2[n-(N-1)]}. \tag{2.38}$$

Indeed, this expression equals the Cramer-Rao bound derived in [5]. However, Papadopoulos and Wornell found experimentally that this bound is not met for large $N-1-n$. Instead, an error threshold exists which depends on the SNR. The actual error variance as a function of N agrees with (2.38) for small $N-1-n$ until levelling out at the error threshold.

To find a more accurate expression for the smoothing error variance, one must account for the fact that $\hat{s}[n] \neq s[n]$. Such an expression is derived below for the case of stationary AWGN and $\beta = 2$.

For notational convenience, let $\mathcal{E}_n$ denote the event that $\hat{s}[n] \neq s[n]$ and let $P_b[n] = \Pr\{\mathcal{E}_n\}$. The following simplifying assumptions allow one to derive an expression for the error threshold which closely agrees with experimental data.

1. Assume that $\mathcal{E}_i$ is independent of i and $\mathcal{E}_j$ for $i \neq j$. Thus, $P_b[n] = P_b$, and one can completely characterize the probability of sign errors by P_b, a steady-state probability.

2. Assume that when $\mathcal{E}_n$ occurs, the estimator produces a fairly good estimate of $-x[n]$ rather than $x[n]$, independent of n, i.e.,

$$\sigma_{\mathcal{E}}^2 \equiv E\left\{(\hat{x}[n|N-1] - x[n])^2|\mathcal{E}_n\right\} \approx E\left\{-4x^2[n]|\mathcal{E}_n\right\}. \tag{2.39}$$

Under these assumptions the smoothing error is given by

$$\begin{aligned}
\sigma^2[n|N-1] &= E\left\{(\hat{x}[n|N-1] - x[n])^2|\bar{\mathcal{E}}_n\bar{\mathcal{E}}_{n+1}\cdots\bar{\mathcal{E}}_{N-2}\right\}(1-P_b)^{N-1-n} \\
&\quad + \sum_{k=0}^{N-2-n} E\left\{(\hat{x}[n|N-1] - x[n])^2|\bar{\mathcal{E}}_n\cdots\bar{\mathcal{E}}_{n+k-1}\mathcal{E}_{n+k}\right\}(1-P_b)^k P_b \\
&= (1-P_b)^{N-1-n}\sigma^2[N-1|N-1]\beta^{-2(N-1-n)} \\
&\quad + \sum_{k=0}^{N-2-n} \sigma_{\mathcal{E}}^2\beta^{-2k}(1-P_b)^k P_b \\
&= (1-P_b)^{N-1-n}\sigma^2[N-1|N-1]\beta^{-2(N-1-n)} \\
&\quad + \sigma_{\mathcal{E}}^2 P_b \frac{1-\beta^{-2(N-1-n)}(1-P_b)^{(N-1-n)}}{1-\beta^{-2}(1-P_b)},
\end{aligned} \tag{2.40}$$

where $\bar{\mathcal{E}}_n$ is the complement of $\mathcal{E}_n$. As expected, (2.40) reduces to the Cramer-Rao bound (2.38) in the limit as P_b approaches zero. The only parameters needed to obtain a more accurate expression in the case of nonzero P_b are P_b and $\sigma_{\mathcal{E}}^2$. These are calculated below.

Steady-State Probability of Sign Error (P_b)

The steady-state probability of sign error P_b is roughly inversely proportional to the square root of SNR, where

$$\text{SNR} \equiv \frac{E\{x^2[n]\}}{\sigma_w^2}. \tag{2.41}$$

One can arrive at this conclusion in the following manner.

Under the same assumptions made in the calculation of the filtering error, one can approximate the residual filtering errors $e[n|n]$ as Gaussian with zero mean and variance $\sigma^2[n|n]$. Empirical results in later sections of this chapter seem to agree with the analytical results which follow from this approximation. Although (2.36) shows that the variance of $e[n|n]$ depends on n, for large n, i.e, in the steady-state,

$$\sigma_e^2 \approx \sigma_w^2(1-\beta^{-2}). \tag{2.42}$$

Assuming that $e[n]$ is Gaussian with zero mean and steady-state variance σ_e^2,

$$P_b[n] = \mathrm{Q}\left(\frac{|x[n]|}{\sigma_e}\right), \tag{2.43}$$

where

$$\mathrm{Q}(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-t^2/2}dt. \tag{2.44}$$

The dependence on n in (2.43) arises from the deterministic view of $x[n]$. If one chooses the stochastic view, one can obtain the average steady-state probability over all tent map sequences parameterized by a given β by integrating (2.43) over the invariant density of $x[n]$. In the case where $\beta = 2$, $x[n]$ is uniform on the interval [-1,1] so by exploiting the even symmetry in (2.43), one finds that

$$\begin{aligned} P_b &= \int \mathrm{Q}\left(\frac{|x|}{\sigma_e}\right) p_{x[n]}(x)dx \\ &= \int_0^1 \mathrm{Q}\left(\frac{x}{\sigma_e}\right) dx \\ &= \sigma_e \int_0^{1/\sigma_e} Q(u)du, \qquad u = x/\sigma_e. \end{aligned} \tag{2.45}$$

The average signal power in this case is 1/3 so that (2.42) can be simplified to $\sigma_e \approx \frac{1}{2\sqrt{\mathrm{SNR}}}$. Also, for SNR $\gg 1$ (0 dB)

$$\int_0^{1/\sigma_e} Q(u)du \approx \int_0^{\infty} Q(u)du = \frac{1}{\sqrt{2\pi}}. \tag{2.46}$$

Thus, by substituting these approximations into (2.45), one can write the steady-state sign error probability as

$$P_b \approx \frac{1}{2}\frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{\mathrm{SNR}}} \approx \frac{0.2}{\sqrt{\mathrm{SNR}}}. \tag{2.47}$$

Hence, the steady-state sign error probability is approximately inversely proportional to the square root of SNR.

Calculation of $\sigma_{\mathcal{E}}^2$

The mean square error in $\hat{x}[n|N-1]$ conditioned on $\mathcal{E}_n$ is roughly inversely proportional to the SNR. One can arrive at this conclusion through the following argument.

Recall, the assumption that the effect of $\mathcal{E}_n$ is to produce a fairly good estimate of $-x[n]$, i.e.,

$$\sigma_{\mathcal{E}}^2 \approx E\{4x^2[n]|\mathcal{E}_n\}. \tag{2.48}$$

Note that $\sigma_{\mathcal{E}}^2 \neq E\{4x^2[n]\}$ since sign errors are more likely to occur when $x^2[n]$ is small than when it is large. Instead,

$$\sigma_{\mathcal{E}}^2 = \int 4x^2 p_{x[n]}(x|\mathcal{E}_n)dx, \tag{2.49}$$

where $p_{x[n]}(x|\mathcal{E}_n)$ is the probability density of $x[n]$ conditioned on $\mathcal{E}_n$.

Bayes' Rule gives the desired density:

$$p_{x[n]}(x|\mathcal{E}_n) = \frac{\Pr\{\mathcal{E}_n|x[n]=x\}p_{x[n]}(x)}{\Pr\{\mathcal{E}_n\}}. \tag{2.50}$$

Equation (2.43) gives $\Pr\{\mathcal{E}_n|x[n]=x\}$, and $\Pr\{\mathcal{E}_n\}$ is P_b by definition. Thus,

$$\sigma_{\mathcal{E}}^2 = \frac{4}{P_b}\int x^2 Q\left(\frac{|x|}{\sigma_e}\right)p_{x[n]}(x)dx. \tag{2.51}$$

Again, in the case $\beta = 2$ one can invoke the approximation $\sigma_e \approx \frac{1}{2\sqrt{\text{SNR}}}$. Also, for SNR $\gg 1$ (0 dB)

$$\int_0^{1/\sigma_e} u^2 Q(u)du \approx \int_0^{\infty} u^2 Q(u)du = \frac{2}{3}\frac{1}{\sqrt{2\pi}} \approx 0.27. \tag{2.52}$$

These approximations, along with (2.47) and (2.51) imply

$$\sigma_{\mathcal{E}}^2 \;=\; \frac{4}{P_b}\int_0^1 x^2 Q\left(\frac{x}{\sigma_e}\right)dx$$

$$\begin{aligned} &= \frac{4}{P_b}\sigma_e^3 \int_0^{1/\sigma_e} u^2 Q(u) du \\ &\approx \frac{2}{3}\frac{1}{\text{SNR}} \\ &\approx \frac{0.67}{\text{SNR}}. \end{aligned} \tag{2.53}$$

Hence, the mean square error in samples where sign errors have occured is roughly inversely proportional to the SNR.

Error Variance Threshold for the ML Estimator

As stated earlier in this chapter, there exists a lower threshold, which will be denoted by σ_{th}^2, for the error variance of the ML estimator for tent map sequences in AWGN. One can find this threshold by taking the limit of (2.40) as $(N-1-n)$ goes to infinity. Taking this limit and assuming $P_b \ll 1$, one obtains

$$\sigma_{\text{th}}^2 \approx \frac{\sigma_\xi^2 P_b}{1-\beta^{-2}}. \tag{2.54}$$

For the case $\beta = 2$, substituting the expressions for P_b and σ_ξ^2 from (2.47) and (2.53) into the above expression for σ_{th}^2 leads to the following result:

$$\sigma_{\text{th}}^2 \approx \frac{4}{9}\frac{1}{\sqrt{2\pi}}\frac{1}{(\text{SNR})^{3/2}} \approx \frac{0.18}{(\text{SNR})^{3/2}}. \tag{2.55}$$

Thus, the error variance lower threshold is inversely proportional to $\text{SNR}^{3/2}$.

2.4 Empirical Simulation Results

To test the expressions for P_b and σ_{th}^2 in (2.47) and (2.55), the sign error probability and error variance threshold were empirically measured from the ML estimates of 1000 sequences of length 50 at SNRs between 0-60 dB. The initial condition $x[0]$ for each of the sequences was randomly generated from a uniform distribution between -1 and 1, consistent with the invariant density for the $\beta = 2$ case. The noise sequences were randomly generated from a Gaussian distribution.

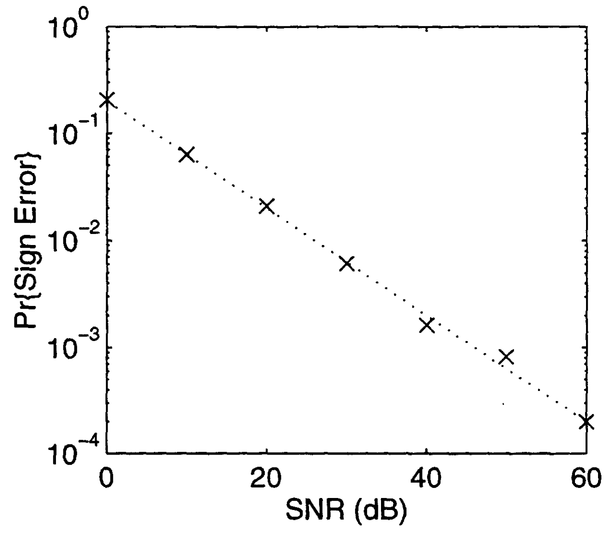

Figure 2-1: Sign Error Probability (P_b). Empirically measured data are marked with "X". Theoretically predicted results are represented by the dotted line.

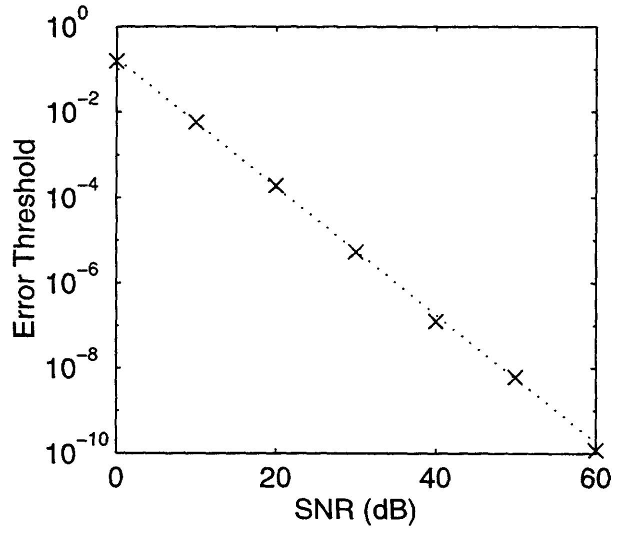

Figure 2-2: Error Variance Threshold (σ^2_{th}). Empirically measured data are marked with "X". Theoretically predicted results are represented by the dotted line.

Figures 2-1 and 2-2 present the results. The dotted lines in each of the figures correspond to (2.47) and (2.55). The empirically measured sign error probabilities are marked with "X" in Figure 2-1 and are taken over all ensembles and times. The error variance threshold, however, is the mean square error averaged only over the first 40 time samples. The last 10 samples were thrown away to remove the transient effects, i.e., the decay of the error variance to the threshold. These empirically measured variance thresholds are marked with "X" in Figure 2-2.

2.5 Comments on the Colored Gaussian Noise Case

The extension of the preceding algorithms and analysis to the colored noise case is less straightforward. In particular, because the $y[n]$ are all dependent, it is not true that $\hat{s}[n|n+m] = \hat{s}[n|n]$ for $m > 0$. Therefore, the estimates $\hat{s}[n|n]$ are not really the signs which should be used in the smoothing stage of the ML estimator (2.13). Instead, the optimal estimator would use $\hat{s}[n|N-1]$. However, if one restricts the estimates of the $s[n]$ to be causal, i.e., $\hat{s}[n] \equiv \hat{s}[n|n]$, then the resulting filter and smoother for the colored Gaussian noise case have the same form as in the white noise case. One still obtains the filtered estimates $\hat{x}[n|n]$ by minimizing $||\mathbf{e}||^2$ in (2.23), but the $\mathbf{S}_n$ matrix is given by

$$\mathbf{S}_n = \operatorname{diag}\left\{\left(-\frac{1}{\beta}\right)^{n-i} \prod_{k=i}^{n-1} \hat{s}[k]\right\} \mathbf{K}_{ww}^{-1} \operatorname{diag}\left\{\left(-\frac{1}{\beta}\right)^{n-j} \prod_{l=j}^{n-1} \hat{s}[l]\right\}. \tag{2.56}$$

The smoothing step (2.13) remains the same. The effect of restricting the estimates of the $s[n]$ to be causal remains an open issue to be explored.

Chapter 3

Tent Map Coding for the AWGN Channel

A wide range of codes have been developed for use with communication channels having fixed and known SNR. There are many instances, however, where such conditions are not met. For instance, on a broadcast channel, SNR typically differs from receiver to receiver so there is no single SNR figure for the system. Likewise, on a time-selective fading channel, SNR is not fixed but rather fluctuates with time. In still other point-to-point channels, the transmitter may not have knowledge of the SNR even when it is fixed. When a channel code is designed for a fixed SNR that is different from the true SNR, the resulting mismatch typically results in performance degradation. If the true SNR is lower than expected, the distortion is higher than expected. Conversely, if the true SNR is higher than expected, the code often does not achieve as small a distortion as that achievable by some other code. This chapter considers the problem of efficient coding for AWGN channels with unknown, multiple, or variable SNRs. More specifically, this chapter explores the potential application of the tent map for joint source-channel coding of analog, discrete-time data.

A traditional digital approach to this problem has been to quantize the source data and encode the quantized data using some suitable channel code so that the quantized data can be recovered with arbitrarily low probability of error. In the absence of channel errors, then, the only distortion results from the quantization.

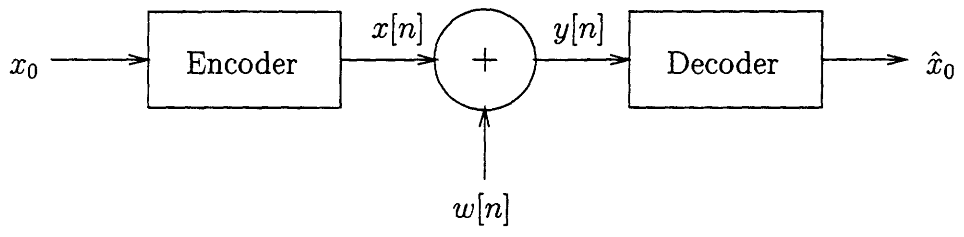

Figure 3-1: Joint Source-Channel Coding of a Uniform Source over an AWGN Channel

However, such an approach, with a finite number of quantization levels, can never achieve arbitrarily small distortion, even when the SNR is arbitrarily large, i.e., such approaches are "quantization error-limited" at high SNR. Thus, for channels with unknown or variable SNR, where the proper number of quantization levels cannot be determined, analog codes or digital-analog hybrids can be used to obtain SNR-limited performance over a broad range of SNR.

This chapter proposes a joint source-channel code based on tent map sequences. Exploiting results from Chapter 2, it is shown that for the AWGN channel and a uniformly distributed source, there exist certain SNRs and bandwidth constraints such that this code always results in a lower mean square error distortion than *any* code with a finite channel alphabet size. Unlike quantization approaches, this tent map code is not quantization error-limited at high SNR.

3.1 Problem Formulation, Coding for the AWGN Channel

Figure 3-1 illustrates the problem considered. A source letter x_0 having a uniform density on the interval $[-1,1]$ is mapped into a sequence $x[n]$ of length N, i.e., the encoder is constrained to expand the bandwidth by a factor of N. The average power P of $x[n]$ is constrained the to be the same as that of an uncoded sequence, i.e.,

$$P = \frac{1}{N}\sum_{n=0}^{N-1} x^2[n] = E\{x_0^2\} = 1/3. \tag{3.1}$$

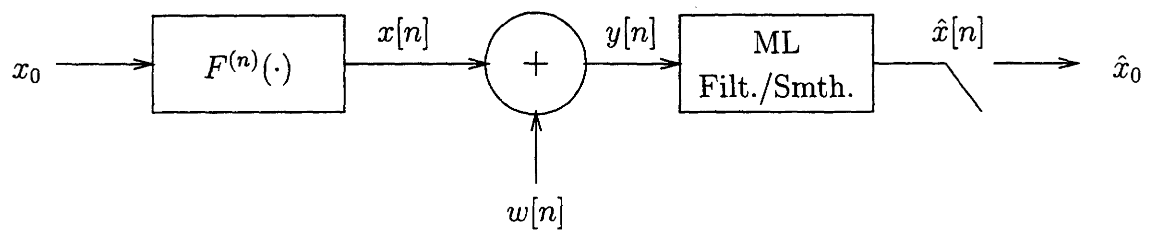

Figure 3-2: Joint Source Channel Coding with the Symmetric Tent Map

This sequence passes through an AWGN channel where the noise $w[n]$ has mean zero and variance σ_w^2. The SNR is defined as in Chapter 2 to be the ratio of the average signal power to the noise variance. Finally, the decoder estimates x_0 from $y[0], \cdots, y[N-1]$, the channel output. The distortion measure is mean square error, $E\{(\hat{x}_0 - x_0)^2\}$. For simplicity, only real-valued baseband channels are considered. Extensions to more typical complex equivalent baseband channels are straightforward. Then, the source-channel coding problem considered is to find a code with small distortion while keeping power and bandwidth fixed.

3.2 Tent Map Twisted Modulation and Coding

This section suggests a method of using the symmetric tent map to address the coding problem described above. The tent map can be used to transmit data over AWGN channels through a method which can be interpreted as a form of twisted modulation [8]. After examining the tent map in this context, it is shown below that a twisted modulation system based on the symmetric tent map is equivalent to a joint source-channel code.

3.2.1 Twisted Modulation

Wozencraft and Jacobs [8] considered the transmission of data through the modulation of a set of orthonormal waveforms. A brief explanation of some of their work is provided below, but only enough to connect their work to the tent map coding system considered in this chapter. For a more detailed explanation of twisted modulation systems, the reader is encouraged to consult [8].

Wozencraft and Jacobs considered transmitting the signal,

$$x[n] = \sum_{i=0}^{N-1} a_i(x_0)\phi_i[n], \tag{3.2}$$

where the set $\{\phi_i[n]\}$ is a set of orthonormal sequences, and the $a_i(x_0)$ represent functions of the source letter. If $x[n]$ is transmitted over a stationary AWGN channel, the receiver observes

$$y[n] = x[n] + w[n]. \tag{3.3}$$

One can represent the sequences $y[n]$, $x[n]$, and $w[n]$ as vectors using the orthonormal set $\{\phi_i[n]\}$ as a basis. Because $w[n]$ is stationary AWGN, the noise vector using this basis is a Gaussian random vector with covariance matrix $\sigma_w^2 I$. Using this basis, one can represent $x[n]$ by the vector

$$\mathbf{x} = [a_0(x_0)\ a_1(x_0) \cdots a_{N-1}(x_0)]^T. \tag{3.4}$$

Wozencraft and Jacobs restrict the class of functions $a_i(x_0)$ to those which have the following property:

$$\left\| \frac{d\mathbf{x}}{dx_0} \right\| = \sqrt{\sum_{i=0}^{N-1} \left(\frac{da_i(x_0)}{dx_0} \right)^2} \equiv \frac{L}{2} \qquad \text{independent of } x_0, \tag{3.5}$$

where $\|\cdot\|$ is the usual Euclidean 2-norm. Under this constraint a differential change in x_0 leads to a proportional change in $\mathbf{x}$, and the constant of proportionality is the same for all x_0. If the $a_i(x_0)$ are linear functions of x_0, the system is called a *linear modulation* system. Otherwise, the system is called a *twisted modulation* system. For example, pulse-position modulation systems are twisted modulation systems, as explained in [8]. The i-fold iterations of the tent map, $F^{(i)}(x_0)$, also satisfy the constraint (3.5), and thus, may modulate the $\phi_i[n]$ in a twisted modulation system with $a_i(x_0) = F^{(i)}(x_0)$. Figure 3-3 shows the locus of possible signal vectors $\mathbf{x}$ for a linear modulation system and for a (tent map) twisted modulation system for the case $N = 2$.

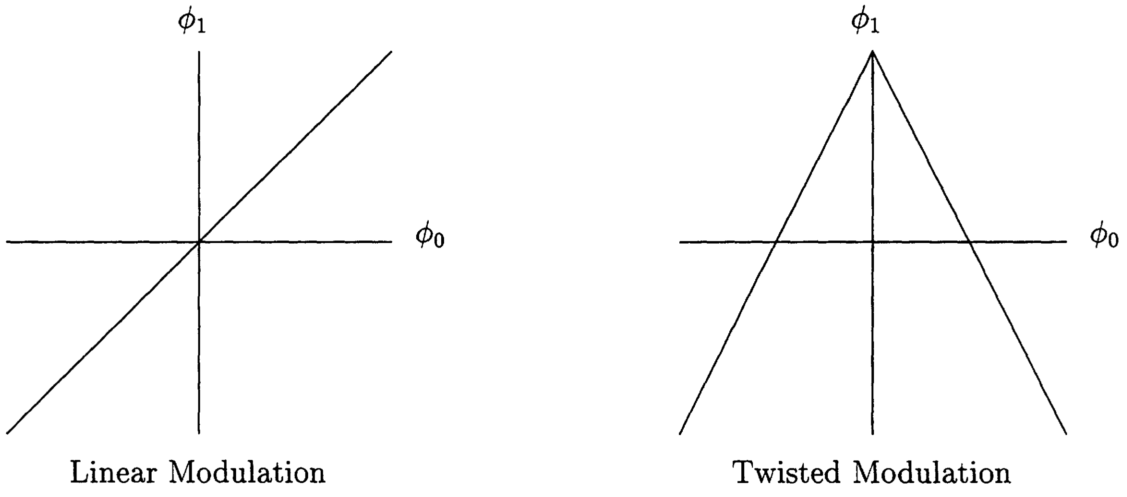

Figure 3-3: Signal Loci for Linear and Twisted Modulation Systems

The maximum likelihood receiver for such systems in stationary AWGN simply projects the received signal vector $\mathbf{y}$ onto the signal locus, and maps the points on the locus uniformly onto the interval $[-1,1]$. From (3.5), the locus will have length L. Thus, for linear modulation systems, mapping onto an interval of length 2 will scale the noise variance by a factor of $\left(\frac{2}{L}\right)^2$. The same result applies for arbitrary twisted modulation systems in high SNR regimes and can be derived by linearization around $\mathbf{x}$ [8]. For the N-dimensional tent map twisted modulation system,

$$\begin{aligned}
\frac{da_i(x_0)}{dx_0} &= \frac{d}{dx_0}F^{(i)}(x_0) \\
&= \frac{d}{dx_0}F\left(F^{(i-1)}(x_0)\right) \\
&= -\beta\frac{d}{dx_0}F^{(i-1)}(x_0) \\
&\vdots \\
&= (-\beta)^i.
\end{aligned} \tag{3.6}$$

Therefore, in the high SNR (weak noise) regime

$$E\{(\hat{x}_0 - x_0)^2\} = \frac{\sigma_w^2}{\left(\frac{L}{2}\right)^2}$$

$$\begin{aligned}
&= \frac{\sigma_w^2}{\sum_{i=0}^{N-1}\left(\frac{da_i(x_0)}{dx_0}\right)^2} \\
&= \frac{\sigma_w^2}{\sum_{i=0}^{N-1}\beta^{2i}} \\
&= \frac{\beta^2-1}{\beta^{2N}-1}\sigma_w^2 \\
&= \beta^{-2(N-1)}\frac{1-\beta^{-2}}{1-\beta^{-2N}}\sigma_w^2. \qquad (3.7)
\end{aligned}$$

This expression is the same as the Cramer-Rao bound discussed in Chapter 2 for the smoothed error variance (2.38) in estimating $x[0]$, the initial value of a tent map sequence.

Equation (3.7) is accurate in the weak noise regime since the signal locus is linear near $\mathbf{x}$. However, as the noise power grows so does the probability that the received vector $\mathbf{y}$ will be closer to a region of the signal locus which does not contain $\mathbf{x}$. In the context of Figure 3-3, this event corresponds to the event that the ϕ_0 components of $\mathbf{y}$ and $\mathbf{x}$ have different signs. Wozencraft and Jacobs call this event a *catastrophic error.* The results in Chapter 2 showed how to compute the actual error variance, taking catastrophic errors into account. Specifically, catastrophic errors lead to a lower threshold on the error variance. An approximation for this threshold is given by (2.55).

3.2.2 Tent Map Coding

From the above discussion one can observe that twisted modulation systems are a class of coding systems. Indeed, one can view the coding problem as defined in Section 3.1 as designing a generalized twisted modulation system, one where the functions $a_i(x_0)$ are not constrained by (3.5). Without the constraint (3.5), however, one would not expect the distortion to be independent of x_0.

Thus, the tent map twisted modulation system described above represents a joint source-channel code. Specifically, the encoder maps x_0 onto the sequence $x[n] = F^{(n)}(x_0)$ where $F^{(n)}(\cdot)$ is the n-fold iteration of the tent map with the parameter $\beta = 2$. As explained in Chapter 1, the code sequence $x[n]$ can be interpreted as representing the quantization of the source x_0. The decoder for the tent map code is the ML estimator described in Chapter 2 followed by a sampler which samples the estimator output at $n = 0$, giving the ML estimate of x_0.

The results from Chapter 2 provide expressions for the distortion resulting from tent map coding for the AWGN channel. In particular, (2.55) implies that for high bandwidths,

$$D_{\text{tent}} \approx \frac{0.18}{\text{SNR}^{3/2}}. \tag{3.8}$$

For low bandwidths the Cramer-Rao bound, or equivalently the weak noise error estimate, provides a good estimate for the distortion,

$$D_{\text{tent}} = \sigma_w^2 \left(\frac{1-\beta^{-2}}{1-\beta^{-2N}}\right)\left(\frac{1}{\beta^2}\right)^{N-1} \leq \sigma_w^2 \left(\frac{1}{\beta^2}\right)^{N-1}, \qquad \beta = 2. \tag{3.9}$$

In practice, D_{tent} is effectively the maximum of (3.8) and (3.9).

3.3 Linear Modulation Codes

Based on the earlier analysis of linear modulation and twisted modulation systems, one can immediately compare the performance of the tent map code with any linear modulation code, i.e., codes which involve modulating a signature sequence of length N with the source letter x_0. The simplest example of a linear modulation code is the repetition code, in which the source letter is simply repeated N times.

The expected distortion for any linear modulation code for the AWGN channel is

$$D_{\text{lin}} = \frac{\sigma_w^2}{N} = \frac{1/3}{N \cdot \text{SNR}}. \tag{3.10}$$

Comparison of (3.8) and (3.9) with (3.10) allows one to find the region in the power-bandwidth plane where the tent map code results in lower distortion than linear modulation codes. In particular, since (3.9) is always less than (3.10) for $N > 1$, the

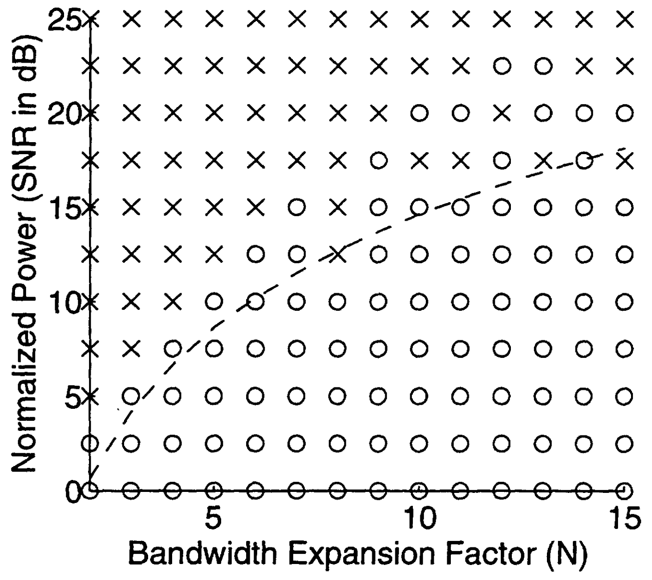

Figure 3-4: Tent Map Coding vs. Linear Modulation Coding. The experimentally determined region in the power-bandwidth plane where the tent map code resulted in lower distortion than the repetition code is marked with X's. The region where the repetition code resulted in lower distortion is marked with O's. The dashed line is the theoretically predicted boundary.

nontrivial condition is that (3.8) be less than (3.10). This condition is equivalent to

$$\mathrm{SNR} \geq 0.29N^2. \tag{3.11}$$

Figure 3-4 shows the region in the power-bandwidth plane where the tent map code results in lower distortion than linear modulation codes. A source sequence of 1000 uniformly distributed random variables on the interval $[-1, 1]$ was encoded using both the tent map and repetition codes with $2 \leq N \leq 15$ and $0 \leq \mathrm{SNR} \leq 25$ dB, and the resulting distortion in the decoded sequence was measured. The dashed curve (– –) represents the theoretically predicted boundary given by (3.11). Looking at the figure, one can see that the tent map code is better than the repetition code in high power or low bandwidth regimes.

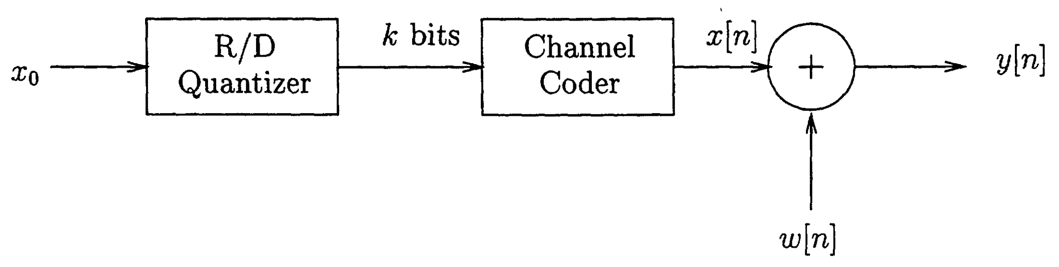

Figure 3-5: Separation of Source and Channel Coding

3.4 An Information Theoretic Perspective

This section examines the tent map code in the context of what is achievable from an information theoretic point of view and shows that there exist power and bandwidth regimes where the tent map code results in smaller distortion than any code with a finite channel alphabet size, e.g., an M-ary code transmitting the quantization of the source. First, optimal coding for unknown SNR channels is considered, along with the short-comings of M-ary coding. Next, the tent map code discussed in Section 3.2 is compared to the best possible M-ary codes. Finally, the tent map code is compared to the best possible (digital or analog) codes.

3.4.1 Optimal Coding of a Uniform Source

In the optimal coding for transmission over a stationary AWGN channel, the separation theorem allows a two-stage cascade of the source encoder and channel encoder without loss of optimality. Thus, the system in Figure 3-5 can perform as well as any joint source-channel coding scheme. The source encoder is a quantizer which represents each source letter with k bits, on average, such that $\frac{k}{N} \leq C$ where C is the capacity of the channel. The channel coder maps the k bits onto a sequence of length N, denoted $x[n]$, which meets the power and bandwidth constraints mentioned before. When an M-ary code is used, $x[n]$ can take one of only M possible values. Finally, $y[n]$ is the channel output. The best possible performance of such a two-stage encoder is also the best possible performance of any joint encoding scheme. More precisely, one can state the separation theorem in the following compact form:

$$R(D) \leq k \leq NC, \tag{3.12}$$

where R(D) is the rate-distortion function and C is the channel capacity. These inequalities can be used to find the minimum possible distortion.

Rate-Distortion Function for a Uniform Source

The rate-distortion function R(D) is the minimum quantization rate needed to achieve an expected distortion of at most D. For the quantizer in Figure 3-5, R(D) is the minimum k such that $E\{(\hat{x}_0 - x_0)^2\} \leq D$ when each source letter is reconstructed from k bits, on average. From rate-distortion theory [2],

$$R(D) = \min_{p(\hat{x}_0|x_0): E\{(\hat{x}_0 - x_0)^2\} \leq D} I(x_0; \hat{x}_0), \tag{3.13}$$

where $p(\hat{x}_0|x_0)$ is the conditional density of $\hat{x}_0$ given x_0, and $I(x_0;\hat{x}_0)$ is the mutual information between x_0 and $\hat{x}_0$. Specifically,

$$I(x_0;\hat{x}_0) \equiv h(x_0) - h(x_0|\hat{x}_0), \tag{3.14}$$

where $h(x_0)$ is the entropy of random variable x_0, and $h(x_0|\hat{x}_0)$ is the conditional entropy of x_0 given $\hat{x}_0$.

A standard method for lower bounding $I(x_0;\hat{x}_0)$ in (3.14) involves defining a new variable $z = x_0 - \hat{x}_0$. After the derivation of the lower bound using this so-called test channel method, an argument showing that the bound is met with equality follows.

By the distortion constraint,

$$E\{z^2\} = E\{(x_0 - \hat{x}_0)^2\} \leq D. \tag{3.15}$$

Since x_0 is uniform on the interval $[-1,1]$, $h(x_0)$ in (3.14) is fixed. In particular,

$$h(x_0) = -\int p_{x_0}(x)\log_2 p_{x_0}(x)dx = 1 \text{ bit}. \tag{3.16}$$

Thus, finding a lower bound on $I(x_0;\hat{x}_0)$ in (3.14) only requires finding an upper bound on $h(x_0|\hat{x}_0)$. The following steps give the desired bound.

$$\begin{aligned} h(x_0|\hat{x}_0) &= h(z+\hat{x}_0|\hat{x}_0) \\ &= h(z|\hat{x}_0) \\ &\leq h(z) \\ &\leq \frac{1}{2}\log_2 2\pi e D. \end{aligned} \tag{3.17}$$

The first line follows from the definition of z. The second follows from the fact that entropy is translation invariant. The third line reflects the fact that conditioning can only decrease entropy. Equality occurs if and only if z is independent of $\hat{x}_0$. The final line is true because the Gaussian density maximizes entropy subject to the second moment constraint in (3.15). Note that the upper bound on $h(x_0|\hat{x}_0)$ given by Equation (3.17) is met with equality if and only if there exists a $p(\hat{x}_0|x_0)$ such that z is Gaussian and independent of $\hat{x}_0$. Such a density may not always exist. Regardless, (3.17) always gives an upper bound and when combined with (3.13), (3.14), and (3.16), gives the desired lower bound on R(D):

$$R(D) \geq 1 - \frac{1}{2}\log_2 2\pi e D = \frac{1}{2}\log_2 \frac{2}{\pi e D}. \tag{3.18}$$

Inverting (3.18) yields the distortion as a function of the number of bits $k = R(D)$.

$$D \geq \frac{2}{\pi e}2^{-2k}. \tag{3.19}$$

The above distortion bound assumes vector quantization, i.e, it is the distortion bound when p source letters are quantized jointly with kp bits in the limit of infinitely large p. With scalar quantization, the distortion increases by a factor of $\frac{\pi e}{6}$ [1] so that the bound becomes

$$D_s \geq \frac{1}{3} 2^{-2k}. \tag{3.20}$$

One can show that (3.20) is met with equality by considering the uniform scalar quantization of x_0 where the quantization regions are all the same size and $\hat{x}_0$ is chosen to be the center of the quantization region. Since x_0 is uniform on the interval $[-1, 1]$ (an interval of length 2) and there are 2^k quantization regions specified by k bits, the length l of each quantization region is

$$l = \frac{2}{2^k}. \tag{3.21}$$

Furthermore, the probability density of x_0 conditioned on the event that x_0 falls within a given region is uniform on that region. Therefore, the expected distortion is equal to the variance of a random variable uniformly distributed on an interval of length l. Specifically,

$$D_s = \frac{1}{3}\left(\frac{l}{2}\right)^2 = \frac{1}{3} 2^{-2k}. \tag{3.22}$$

Thus, (3.20) is met with equality. Since vector quantization decreases the distortion by $\frac{\pi e}{6}$, (3.19) and (3.18) are met with equality also. Thus, recalling (3.12), one can determine that the optimal code results in a distortion of

$$D = \frac{2}{\pi e} 2^{-2k} = \frac{2}{\pi e} (2^2)^{-NC}. \tag{3.23}$$

Note that (3.23) depends only on the channel capacity.

Optimal Coding for the Unknown SNR Channel

The capacity of an AWGN channel is well-known to be

$$C_{\text{AWGN}} = \frac{1}{2} \log_2(1 + \text{SNR}). \tag{3.24}$$

If the SNR is known at the time of transmission, one can determine C and, thus, determine k. However, if the SNR is variable, an optimal code must still transmit

NC bits in the quantization of each source letter, where C varies with the SNR. Each source letter can be represented in the following way:

$$x_0 \leftrightarrow s[0], s[1], \cdots, s[k_{\mathrm{SNR}}-1], \cdots, s[k_{\max}-1], \tag{3.25}$$

where $s[i-1]$ is the ith most significant bit in quantization of x_0 and $k_{\max} = NC_{\max}$ is the number of bits per source letter which can be transmitted reliably when the SNR is $\mathrm{SNR}_{\max}$, the maximum possible SNR. Similarly, k_{SNR} is the number of bits per source letter which can be transmitted reliably at the actual SNR. An optimal code must have the following two properties.

1. All of the bits in (3.25) must be recoverable from the (uncorrupted) transmitted sequence. Otherwise, the code would be suboptimal when $\mathrm{SNR} = \mathrm{SNR}_{\max}$.

2. At the actual SNR, the k_{SNR} most significant bits must be recoverable from the (corrupted) transmitted sequence.

A code satisfying these properties may be difficult to find, especially if $\mathrm{SNR}_{\max}$ is infinite or unknown, and it will be shown below that when the $\mathrm{SNR}_{\max}$ is infinite, the optimal channel code will not be purely digital (M-ary). Furthermore, even if the SNR is known and a purely digital optimal code for that SNR exists, one can construct a digital-analog hybrid which is more robust to uncertainties in the SNR.

Claim 1 *If the* $\mathrm{SNR}_{\max}$ *is infinite, no optimal purely digital code exists.*

Proof. A purely digital code, by definition, draws from a finite alphabet of size M. The channel capacity when the channel input is restricted to one of M levels is upper bounded by

$$\begin{aligned} C_M &= \max I(X;Y) \\ &= \max H(X) - H(X|Y) \\ &\le \max H(X) \\ &= \log_2 M \end{aligned} \tag{3.26}$$

with equality if and only if the input can be determined from the output with no uncertainty. Then, at most, an average of $N \log_2 M$ bits can be transmitted per source letter. But, $k_{\max}$ is infinite when $\mathrm{SNR}_{\max}$ is infinite. Thus, since M and N are finite, a purely digital code can not be optimal.

□

The difficulty in applying M-ary coding to channels with unknown SNR lies in determining the proper k, the number of bits per source letter in the quantization. The symmetric tent map provides a means to transform an M-ary code which is optimal at some fixed SNR into a digital-analog hybrid code which sends *all* the bits in the quantization of each source letter. This hybrid code is more robust to uncertainties in the SNR. An M-ary code which is optimal at a known, fixed $\mathrm{SNR}_{\mathrm{opt}}$ can transmit reliably, on average, the k_{opt} most significant bits of a source letter in N uses of the channel, where

$$k_{\mathrm{opt}} = N\frac{1}{2}\log_2(1 + \mathrm{SNR}_{\mathrm{opt}}). \tag{3.27}$$

One can construct a digital-analog hybrid in the following way:

1. The optimal digital code sends the first $k_{\mathrm{dig}} = k_{\mathrm{opt}}(N-1)/N$ bits of the quantization of x_0 in the first $N-1$ channel uses per source letter.

2. The Nth channel use per source letter is used to send the $(k_{\mathrm{dig}}+1)$th element of the tent map sequence generated from x_0, i.e., $x[N-1] = F^{(k_{\mathrm{dig}})}(x_0)$. This real number contains information about bits $s[k_{\mathrm{dig}}], s[k_{\mathrm{dig}}+1], \cdots$.

Then, one can make the following statements about the performance of this digital-analog hybrid.

Claim 2 *The digital-analog hybrid code performs better than the M-ary code with vector quantization if the* SNR *is greater than* $(\mathrm{SNR}_{\mathrm{opt}}+1)\frac{\pi e}{6}$. *The digital-analog hybrid code performs better than the M-ary code with scalar quantization if the SNR is greater than* $(\mathrm{SNR}_{\mathrm{opt}}+1)$.

Proof. Since SNR > SNR_{opt}, the hybrid code transmits the first k_{dig} most significant bits in the quantization of each source letter with arbitrarily low probability of error. Then, the decoder is

$$\hat{x}_0 = F^{(-k_{\text{dig}})}_{s[0],s[1],\cdots,s[k_{\text{dig}}-1]}(y[N-1]), \tag{3.28}$$

and the distortion is

$$\begin{aligned} D_{\text{hyb}} &= \sigma_w^2 \left(\frac{1}{\beta^2}\right)^{k_{\text{dig}}} \\ &= \sigma_w^2 \left(\frac{1}{\beta^2}\right)^{k_{\text{opt}}(N-1)/N} \\ &= P\frac{\sigma_w^2}{P}\left(\frac{1}{2^{k_{\text{opt}}}2^{-k_{\text{opt}}/N}}\right)^2 \\ &= \frac{1}{3}2^{-2k_{\text{opt}}}\frac{1+\text{SNR}_{\text{opt}}}{\text{SNR}}, \end{aligned} \tag{3.29}$$

where the last line makes use of (3.27). From (3.23), the distortion of the digital code for SNR $\geq \text{SNR}_{\text{opt}}$ is

$$D_{\text{dig}} = \frac{2}{\pi e}2^{-2k_{\text{opt}}}. \tag{3.30}$$

Thus, the digital-analog hybrid is better than the digital code whenever SNR > $(\text{SNR}_{\text{opt}}+1)\frac{\pi e}{6}$ so the claim is proved. Also, if only scalar quantization is allowed, the distortion of the digital code increases by a factor of $\frac{\pi e}{6}$ so the hybrid code becomes better at SNR > $\text{SNR}_{\text{opt}} + 1$.

□

Corollary 1 *When the* SNR *is exactly* SNR_{opt}*, the hybrid code is worse than the optimal digital code, but is only worse by* $\frac{\pi e}{6} \simeq 1.53$ dB *at high* SNR_{opt} *when vector quantization is allowed. Furthermore, when only scalar quantization is allowed, the hybrid code is asymptotically as good as the optimal digital code at high* SNR_{opt}.

Proof. From the proof of Claim 2, when only scalar quantization of the source is allowed and the SNR is SNR_{opt},

$$\lim_{\text{SNR}_{\text{opt}}\to\infty}\frac{D_{\text{hyb}}}{D_{\text{dig}}} = \lim_{\text{SNR}_{\text{opt}}\to\infty} 1 + \frac{1}{\text{SNR}_{\text{opt}}} = 1. \tag{3.31}$$

Thus, the hybrid code is asymptotically as good as the digital code in the case of scalar quantization. If vector quantization is allowed, the hybrid code is worse by only $\frac{\pi e}{6} = 1.53$ dB at high $\mathrm{SNR}_{\mathrm{opt}}$.

□

Finally, the two codes can be expected to perform similarly at SNRs much lower than $\mathrm{SNR}_{\mathrm{opt}}$ since the most significant bits are encoded with the same (digital) channel code. The distortion will be determined by how well these bits can be recovered.

3.4.2 Tent Map vs. M-ary Coding

Having seen the short comings of purely digital coding for channels with unknown or variable SNR, one might expect that analog codes (such as the tent map code in Section 3.2) may perform better in certain scenarios. This section compares the expected distortion resulting from the tent map code to that of the smallest achievable distortion using a channel code with a finite alphabet size M. For any finite M there always exists some power and bandwidth such that the tent map code yields smaller expected distortion. Sufficient conditions on SNR and N under which this statement is true are derived below.

Combining the upper bound for the M-ary channel capacity (3.26) with (3.23) yields a lower bound on the distortion of the best possible M-ary code,

$$D_M \geq \frac{2}{\pi e}(M^2)^{-N}. \tag{3.32}$$

This bound is met with equality only at infinitely high SNR where the channel input can be resolved with zero probability of error. Comparison of (3.8) and (3.9) with (3.32) yields sufficient (but not necessary) conditions under which the tent map code is guaranteed to result in smaller expected distortion than any M-ary code:

$$\mathrm{SNR} \geq (-0.762 + 10N\frac{2}{3}\log_{10} M^2) \quad \mathrm{dB} \tag{3.33}$$

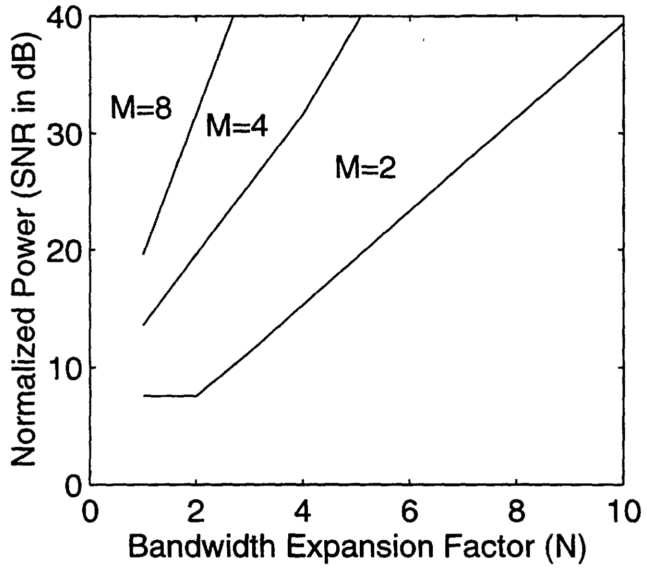

Figure 3-6: Tent Map Coding vs. M-ary Coding. Regions in power-bandwidth plane where the tent map code results in lower expected distortion than any M-ary code. The regions in which the tent map code results in lower distortion lie above the curves, i.e., high SNR and low bandwidth.

and

$$\text{SNR} \geq (7.55 + 10N \log_{10} \frac{M^2}{4}) \quad \text{dB}. \tag{3.34}$$

The conditions (3.33) and (3.34) define a region in the power-bandwidth plane. In regions corresponding to high power (SNR) and low bandwidth (N), tent map coding results in smaller distortion than any M-ary code. The boundaries for the regions corresponding to $M = 2$, $M = 4$, and $M = 8$ are plotted in Figure 3-6.

3.4.3 Distortion Bounds for the AWGN Channel

This section examines the performance of the tent map code in the context of what is achievable by any code, digital or analog. The distortion bound (3.23) expresses the smallest possible distortion as a function of the channel capacity, given by (3.24) in the case of an AWGN channel. Thus, by examining how various coding strategies constrain channel capacity, one can gain some insight into the potential performance of such strategies.

The symmetric tent map code produces a channel input with the same density as the source, namely, uniform on the interval $[-1,1]$. The channel capacity when the input is constrained to have a uniform distribution is

$$\begin{aligned} C_{\text{UNI}} &= I(X;Y) \\ &= h(Y) - h(Y|X) \\ &= h(Y) - h(W) \\ &= h(Y) - \frac{1}{2}\log_2 2\pi e \sigma_w^2 \\ &= -\int p_Y(y)\log_2 p_Y(y)dy - \frac{1}{2}\log_2 2\pi e\sigma_w^2 \end{aligned} \tag{3.35}$$

where $p_Y(y)$ is the convolution of a uniform density with a Gaussian density, viz.,

$$p_Y(y) = \frac{1}{2}\left[Q\left(\frac{y-1}{\sqrt{\sigma_w^2}}\right) - Q\left(\frac{y+1}{\sqrt{\sigma_w^2}}\right)\right] \tag{3.36}$$

The distortion bounds for the unconstrained and uniform input constrained stationary AWGN channel, along with the distortion curve for the tent map code, are plotted in Figures 3-7 and 3-8 for SNRs of 20 and 30 dB, respectively. The distortion curves for the repetition code and the best possible binary code are included also. For the binary code ($M = 2$), the exact capacity is given by

$$C_{\text{BSC}} = 1 - H(\epsilon), \qquad \epsilon = Q\left(\sqrt{\text{SNR}}\right) \tag{3.37}$$

Note that the bounds for the unconstrained AWGN channel and the uniform input constrained AWGN channel are quite close. The decrease in channel capacity resulting from the uniform input restriction is quite small. Thus, the uniform channel input constraint inherent in the use of the tent map code does not limit performance to any significant extent. Also, at high bandwidths this tent map code does not trade-off bandwidth for smaller distortion. Papadopoulos and Wornell explain this threshold behavior as resulting from the sensitive dependence characteristics of chaotic dynamics [5]. The digital-analog hybrid code mentioned earlier partially addresses this issue, although finding a pure tent map code or other nonlinear code which is bandwidth

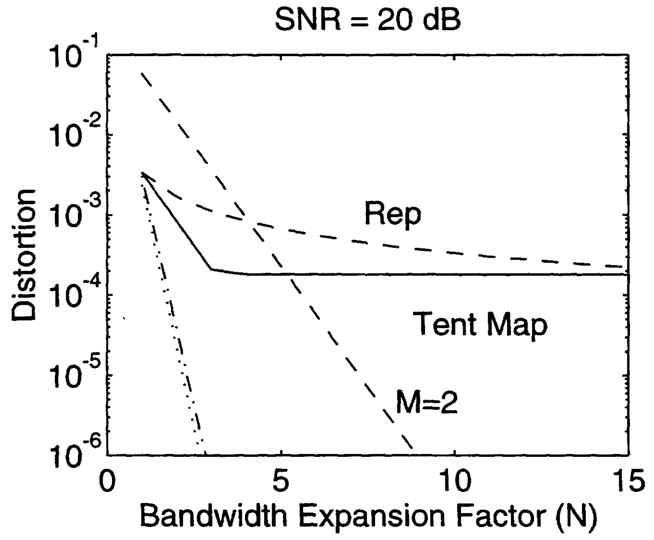

Figure 3-7: Distortion Bounds at SNR of 20 dB. The dotted (···) curve corresponds to the unconstrained AWGN channel, and the dash-dot (— · —) curve corresponds to the uniform input constrained AWGN channel.

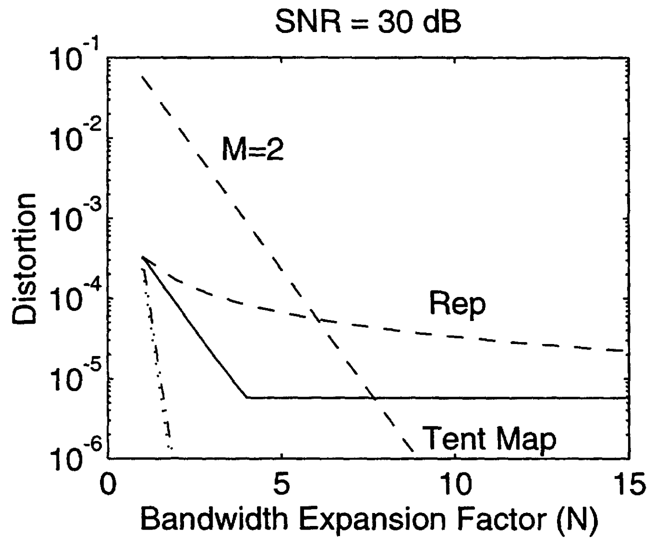

Figure 3-8: Distortion Bounds at SNR of 30 dB. The dotted (···) curve corresponds to the unconstrained AWGN channel, and the dash-dot (— · —) curve corresponds to the uniform input constrained AWGN channel.

scalable remains an open problem.

Chapter 4

Deconvolution of Tent Map Signals

Previous chapters considered channels which only added noise to the input signal. However, dispersion in the form of convolutional distortion is often encountered as well. These channels cause intersymbol interference (ISI) which must be removed through a deconvolution/equalization process. This chapter deals with the removal of ISI when chaotic signals are transmitted over unknown ISI channels. The chaotic signal may be a signal used to probe the channel, for example, or the chaotic signal may be the signal of interest which one wishes to recover.

Figure 4-1 illustrates a proposed structure for the estimation of a tent map sequence in the presence of ISI. The tent map signal $x[n]$ is convolved with the unknown channel impulse response $h[n]$. The received signal $r[n]$ is a noisy version of the channel output. A linear equalizer, a filter with impulse response $g[n]$, produces an estimate of $x[n]$, denoted $\tilde{x}[n]$. For reasons such as simplicity of implementation, the

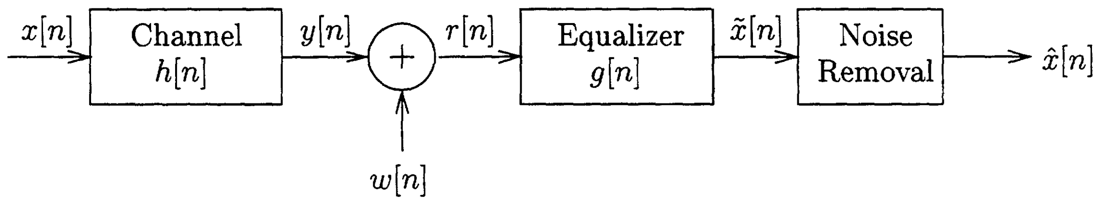

Figure 4-1: Convolutional Distortion and Equalization

equalizer is constrained to be an FIR filter. Further processing attempts to remove residual noise from $\tilde{x}[n]$. The approach taken in this thesis uses the ML estimator in AWGN for noise removal, although this approach may be suboptimal since there is no reason to believe that $\tilde{x}[n]$ is well-modeled by $x[n]$ plus AWGN. The remainder of this chapter explores three algorithms for choosing the filter taps of the M-tap equalizer when the channel input is a tent map sequence of length N.

4.1 Deconvolution Algorithms

This section derives the three algorithms for selecting the equalizer taps. The first two algorithms make explicit use of the known, deterministic structure of chaotic signals, while the third algorithm exploits the sensitivity to initial condition characteristic of chaos.

4.1.1 The Dynamics Matching Algorithm

This section describes an algorithm for selecting the equalizer taps based on a criteria suggested by Isabelle [4]. Although this chapter focuses on the deconvolution of tent map signals, the dynamics matching algorithm can be applied more generally to sequences whose dynamics correspond to a piecewise linear map. In particular, the algorithm can be applied whenever $x[n]$ in Figure 4-1 has the following form:

$$x[n] = F(x[n-1]), \tag{4.1}$$

where

$$F(x) = m_{s(x)}x + b_{s(x)}. \tag{4.2}$$

Each linear region of the map is known as a symbol, and the sequence $x[n]$ may be represented by the sequence of symbols $s[n]$, the sequence of regions in each of the $x[n]$ fall. Such a representation is known as the symbolic dynamics representation. The $m_{s(x)}$ and $b_{s(x)}$ in (4.2) represent the slope and y-intercept of the line in symbol s. The tent map is a piecewise linear map with $s = \text{sgn}(x)$, $m_{-1} = \beta$, $m_1 = -\beta$, and

$b_s = \beta - 1.$

Isabelle's Criteria

The dynamics matching algorithm selects the equalizer taps

$$\hat{\mathbf{g}} = [\hat{g}[0]\ \hat{g}[1]\ \cdots\ \hat{g}[M-1]]^T \tag{4.3}$$

based on the dynamics matching criteria suggested by Isabelle. Isabelle has shown that an FIR filter whose input and output both follow the same map is a simple delay [4]. He conjectures, then, that any (IIR or FIR) LTI filter for which the input and output maps are identical must be a simple delay. Because the channel and equalizer in cascade act as a single LTI filter, the appropriate equalizer is the one whose output obeys the same map as the input. Therefore, the desired equalizer produces an output such that

$$\tilde{x}[n] = F(\tilde{x}[n-1]). \tag{4.4}$$

Solving (4.4) to find the equalizer taps is known as dynamics matching.

Of course, due to noise and non-invertibility of the channel, (4.4) may not be solvable exactly. Isabelle suggests finding the least-squares solution to (4.4). In particular, Isabelle's dynamics matching criteria is to select the $\hat{\mathbf{g}}$ such that

$$\hat{\mathbf{g}} = \arg\min_{\mathbf{g}} J(\mathbf{g}) = \arg\min_{\mathbf{g}} \sum_{n=1}^{N-1} \left(\tilde{x}[n] - F(\tilde{x}[n-1])\right)^2, \tag{4.5}$$

where

$$\begin{aligned} \tilde{x}[n] &= \sum_{k=0}^{M-1} g[k]r[n-k] && (4.6) \\ \tilde{x}[n-1] &= \sum_{k=0}^{M-1} g[k]r[n-1-k]. && (4.7) \end{aligned}$$

The Algorithm

This section derives an algorithm based on Isabelle's dynamics matching criteria described above. Although the algorithm is not guaranteed to find the least squares solution to (4.4), it finds a solution which meets a necessary condition on the least squares solution to (4.4), provided the algorithm converges.

By substituting (4.6) and (4.7) into (4.4), recalling (4.2), and rearranging terms, one can show that

$$\sum_{k=0}^{M-1} g[k]\left(r[n-k] - m_{s[n-1]}r[n-1-k]\right) = b_{s[n-1]} \tag{4.8}$$

$$1 \le n \le N-1, \tag{4.9}$$

where $s[n-1]$ is the symbol associated with $\tilde{x}[n-1]$. One can write (4.8) as the matrix equation,

$$\mathbf{R}_{\mathbf{s}(\mathbf{g})}\mathbf{g} = \mathbf{b}_{\mathbf{s}(\mathbf{g})}, \tag{4.10}$$

where for $1 \le i \le N-1$ and $1 \le j \le M$

$$[\mathbf{R}_\mathbf{s}]_{ij} = r[i-(j-1)] - m_{s[i-1]}r[i-1-(j-1)], \tag{4.11}$$

$$[\mathbf{b}_\mathbf{s}]_i = b_{s[i-1]}, \tag{4.12}$$

$$[\mathbf{g}]_j = g[j-1] \tag{4.13}$$

and where the vector

$$\mathbf{s} = [s[0] \ \cdots \ s[N-2]]^T \tag{4.14}$$

is a function of g through (4.6) and (4.7). If one is given an estimate of s, denoted by $\hat{\mathbf{s}}$, then the least squares solution to (4.10) given $\hat{\mathbf{s}}$ is

$$\hat{\mathbf{g}}_{\hat{\mathbf{s}}} = (\mathbf{R}_{\hat{\mathbf{s}}}{}^T\mathbf{R}_{\hat{\mathbf{s}}})^{-1}\mathbf{R}_{\hat{\mathbf{s}}}{}^T\mathbf{b}_{\hat{\mathbf{s}}}. \tag{4.15}$$

This solution corresponds to the minimum of the objective function

$$J_{\hat{\mathbf{s}}}(\mathbf{g}) = \sum_{n=1}^{N-1} \left(\tilde{x}[n] - F_{\hat{\mathbf{s}}[n-1]}(\tilde{x}[n-1])\right)^2, \tag{4.16}$$

where

$$F_s(x) = m_s x + b_s \tag{4.17}$$

so that $F_s(x) = F(x)$ when $s = s(x)$. One should note that the solution defined by (4.15) is not necessarily a global minimum of J in (4.5) since

$$(\hat{\mathbf{g}}, \mathbf{s}) = \left(\arg\min_{\mathbf{g}} J(\mathbf{g}), \mathbf{s}(\mathbf{g})\right) = \arg\min_{(\mathbf{g},\mathbf{s}):\mathbf{s}=\mathbf{s}(\mathbf{g})} J_{\mathbf{s}}(\mathbf{g}), \tag{4.18}$$

and $\hat{\mathbf{g}}_{\hat{\mathbf{s}}}$ minimizes $J_{\mathbf{s}}(\mathbf{g})$ only for a particular $\mathbf{s} = \hat{\mathbf{s}}$. Also, one does not know that $\hat{\mathbf{s}} = \mathbf{s}(\hat{\mathbf{g}}_{\hat{\mathbf{s}}})$. A necessary condition for $\hat{\mathbf{g}}_{\hat{\mathbf{s}}} = \hat{\mathbf{g}}$ is that

$$\hat{\mathbf{g}}_{\hat{\mathbf{s}}} = \arg\min_{\mathbf{g}:\mathbf{s}(\mathbf{g})=\mathbf{s}(\hat{\mathbf{g}}_{\hat{\mathbf{s}}})} J_{\mathbf{s}(\hat{\mathbf{g}}_{\hat{\mathbf{s}}})}(\mathbf{g}). \tag{4.19}$$

This necessary condition suggests the following iteration, which will be referred to as the dynamics matching algorithm,

1. Guess a solution $\hat{\mathbf{g}}^{(i-1)}$.

2. Calculate $\tilde{x}[n-1]$ from (4.7) for $1 \le n \le N-1$ to find $\hat{\mathbf{s}} = \mathbf{s}(\hat{\mathbf{g}}^{(i-1)})$ and the resulting $\mathbf{R}_{\hat{\mathbf{s}}}$ and $\mathbf{b}_{\hat{\mathbf{s}}}$.

3. Find a new solution $\hat{\mathbf{g}}^{(i)}$ from (4.15). Iterate between steps 2 and 3 until $\hat{\mathbf{g}}^{(i-1)} = \hat{\mathbf{g}}^{(i)}$.

If the above algorithm converges, then

$$\hat{\mathbf{g}}^{(i)} = \arg\min_{\mathbf{g}} J_{\mathbf{s}(\hat{\mathbf{g}}^{(i)})}(\mathbf{g}). \tag{4.20}$$

This condition is sufficient to guarantee that $\hat{\mathbf{g}}^{(i)}$ meets the necessary condition (4.19).

A Recursive Implementation

The block dynamics matching algorithm described above produces outputs only after receiving entire blocks of data, i.e., the entire tent map sequence. A recursive implementation of the dynamics matching algorithm is developed below which produces a new sample of output each time a new input sample is received.

Suppose the set of taps

$$\hat{\mathbf{g}}_{N-1} = [\hat{g}[0|N-1] \ \cdots \ \hat{g}[M-1|N-1]]^T \tag{4.21}$$

represents a "good" equalizer based on observation of $r[0], \cdots, r[N-1]$ in the sense that it is the least squares solution to

$$\mathbf{R}_{\hat{\mathbf{s}}_{N-1}} \mathbf{g} = \mathbf{b}_{\hat{\mathbf{s}}_{N-1}}, \tag{4.22}$$

where

$$\hat{\mathbf{s}}_{N-1} = [s[0|0] \ s[1|1] \ \cdots \ s[N-2|N-2]]^T \tag{4.23}$$

and where $s[n|m]$ is the symbol corresponding to the $\tilde{x}[n]$ computed from the filter taps $\hat{\mathbf{g}}_m$, i.e.,

$$s[n|m] = \text{symbol}\left(\sum_{k=0}^{M-1} \hat{g}[k|m] r[n-1-k]\right). \tag{4.24}$$

The recursive deconvolution problem is to refine the equalizer taps after receiving a new data sample $r[N]$.

The new data sample defines a new row of $\mathbf{R}$ and $\mathbf{b}$, denoted $\mathbf{r}_N^T$ and b_N, i.e.,

$$[\mathbf{r}_N^T]_j = r[N-(j-1)] - m_{s[N-1|N-1]} r[N-1-(j-1)] \tag{4.25}$$

$$b_N = b_{s[N-1|N-1]}. \tag{4.26}$$

Therefore, a reasonable way to refine the equalizer taps is to select the least squares solution to

$$\underbrace{\begin{bmatrix} \mathbf{R}_{\hat{\mathbf{s}}_{N-1}} \\ \mathbf{r}_N^T \end{bmatrix}}_{\mathbf{R}_{\hat{\mathbf{s}}_N}} \mathbf{g} = \underbrace{\begin{bmatrix} \mathbf{b}_{\hat{\mathbf{s}}_{N-1}} \\ b_N \end{bmatrix}}_{\mathbf{b}_{\hat{\mathbf{s}}_N}}. \tag{4.27}$$

Equation (4.15) in turn gives the solution for the equalizer taps. One can update $(\mathbf{R}_{\hat{\mathbf{s}}}^T \mathbf{R}_{\hat{\mathbf{s}}})^{-1}$ and $\mathbf{R}_{\hat{\mathbf{s}}}^T \mathbf{b}_{\hat{\mathbf{s}}}$ recursively via

$$\mathbf{R}_{\hat{\mathbf{s}}_N}^T \mathbf{R}_{\hat{\mathbf{s}}_N} = \mathbf{R}_{\hat{\mathbf{s}}_{N-1}}^T \mathbf{R}_{\hat{\mathbf{s}}_{N-1}} + \mathbf{r}_N \mathbf{r}_N^T \tag{4.28}$$

and

$$\mathbf{R}_{\hat{\mathbf{s}}_N}^T \mathbf{b}_{\hat{\mathbf{s}}_N} = \mathbf{R}_{\hat{\mathbf{s}}_{N-1}}^T \mathbf{b}_{\hat{\mathbf{s}}_{N-1}} + \mathbf{r}_N b_N. \tag{4.29}$$

Thus, (4.28) and (4.29), along with (4.15), give the recursive implementation of the dynamics matching algorithm. With this implementation, one needs only to store $\mathbf{R}_{\hat{\mathbf{s}}_N}^T \mathbf{R}_{\hat{\mathbf{s}}_N}$ (an $M \times M$ matrix) and $\mathbf{R}_{\hat{\mathbf{s}}_N}^T \mathbf{b}_{\hat{\mathbf{s}}_N}$ (an $M \times 1$ vector) rather than $\mathbf{R}_{\hat{\mathbf{s}}_N}$ (an $N \times M$ matrix). Other well-known methods to recursively solve (4.27) exist. Some of these can be found in [3].

4.1.2 The Alternating Projections Algorithm

The alternating projections algorithm selects the equalizer taps to minimize the mean square distance between the equalizer output and the closest tent map sequence of length N. It performs the minimization by alternately projecting onto the set of equalizer outputs and onto the set of tent map sequences.

If $\mathcal{T}$ is the set of all tent map sequences, then the channel input is in $\mathcal{T}$,

$$\mathbf{x}_{\text{true}} \in \mathcal{T} = \{\mathbf{x} : [\mathbf{x}]_i = F([\mathbf{x}]_{i-1})\}. \tag{4.30}$$

The equalizer output set ($\mathcal{O}$) is the set of all possible outputs from an M-tap equalizer given the equalizer input $r[n]$,

$$\mathcal{O} = \left\{\tilde{\mathbf{x}} : \tilde{\mathbf{x}} = \mathbf{R}\mathbf{g},\ \mathbf{g} \in \mathcal{R}^M\right\}, \tag{4.31}$$

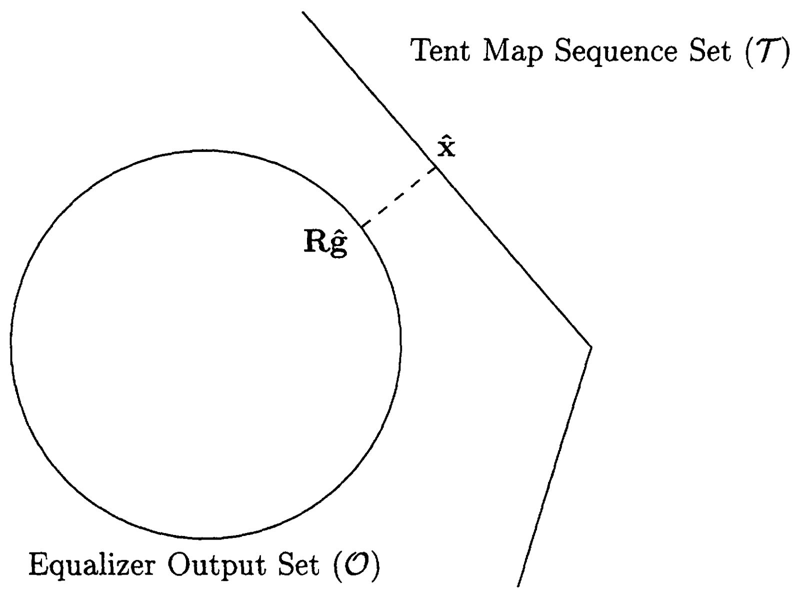
Tent Map Sequence Set ($\mathcal{T}$)
$\hat{\mathbf{x}}$
$\mathbf{R}\hat{\mathbf{g}}$
Equalizer Output Set ($\mathcal{O}$)

Figure 4-2: Selection Criterion of the Alternating Projections Algorithm. The goal is to minimize the mean square distance between the equalizer output and the closest tent map sequence.

where

$$[\mathbf{R}]_{ij} = r[i-j] \tag{4.32}$$

and therefore, $\mathbf{Rg}$ is the convolution of $r[n]$ and $g[n]$.

The goal of the alternating projections algorithm is to jointly estimate the channel input and equalizer taps such that

$$(\hat{\mathbf{x}}, \hat{\mathbf{g}}) = \underset{(\mathbf{x},\mathbf{g}):\mathbf{x}\in\mathcal{T}}{\arg\min} \|\mathbf{x} - \mathbf{Rg}\|^2, \tag{4.33}$$

where $\|\cdot\|$ is the usual Euclidean 2-norm. This selection criterion is depicted in Figure 4-2.

The alternating projections algorithm iteratively performs the minimization described above, alternating between minimization over $\mathbf{x}$ and over $\mathbf{g}$.

1. Guess a set of equalizer taps $\hat{\mathbf{g}}^{(i-1)}$.

2. Find the tent map sequence closest in a least squares sense to $\mathbf{R}\hat{\mathbf{g}}^{(i-1)}$,

$$\hat{\mathbf{x}}^{(i)} = \arg\min_{\mathbf{x}:\mathbf{x}\in\mathcal{T}} \|\mathbf{x} - \mathbf{R}\hat{\mathbf{g}}^{(i-1)}\|^2. \tag{4.34}$$

3. Project $\hat{\mathbf{x}}^{(i)}$ onto the column space of $\mathbf{R}$ to find $\hat{\mathbf{g}}^{(i)}$,

$$\hat{\mathbf{g}}^{(i)} = \arg\min_{\mathbf{g}} \|\hat{\mathbf{x}}^{(i)} - \mathbf{R}\mathbf{g}\|^2 = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\hat{\mathbf{x}}^{(i)}. \tag{4.35}$$

4. Increment i and return to step 2. Continue iterating until $|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(i-1)}| < \epsilon$, where ϵ is some termination tolerance.

This algorithm has several convenient features. First, the iterative minimization described above converges monotonically to a local minimum, i.e., each iteration finds a pair of sequences which are at least as close as the previous pair. The proof of this statement is given in Section 4.2.

The second convenient feature is the ML estimator from Chapter 2 for tent map sequences in AWGN can perform the minimization specified in step 2. Step 2 is equivalent to finding the ML estimate of $\hat{\mathbf{x}}$ when $\mathbf{R}\hat{\mathbf{g}}^{(i-1)}$ is $\hat{\mathbf{x}}$ plus AWGN. Thus, a computationally efficient minimization routine exists.

Finally, note that the matrix inversion in step 3, the only step where the number of computations depends cubically on the number of equalizer taps, does not need to be performed more than once since the matrix $\mathbf{R}$ does not change with each iteration.

4.1.3 The Supex Algorithm

The supex algorithm, developed by Shalvi and Weinstein [6], finds the equalizer taps when the channel input satisfies certain second and fourth-order whiteness constraints. For example, sequences of independent random variables are guaranteed to satisfy these constraints. With each iteration of the supex algorithm, the impulse response of the cascaded channel and equalizer approaches a delay. Specifically, the largest tap of the cascaded system grows exponentially relative to all the other taps.

Clearly, neither tent map sequences in particular, nor chaotic sequences in general, are sequences of independent random variables since each element is a deterministic function of the previous element. However, given the sensitivity to initial condition characteristic of chaos, one might expect that downsampling chaotic sequences makes them more amenable to equalization using the supex algorithm. Indeed, a downsampled tent map sequence is more white than the original sequence. Thus, a downsampler is an approximate whitener for these sequences.

Whitening of Tent Map Sequences through Downsampling

The simplest implementation of the supex algorithm requires that $x[n]$ satisfies certain second and fourth-order whiteness properties. For zero-mean random variables, the second and fourth-order cumulants are defined as

$$\mathrm{cum}(x_1; x_2) = E\{x_1 x_2\} \tag{4.36}$$

and

$$\begin{aligned}\mathrm{cum}(x_1; x_2; x_3; x_4) &= E\{x_1 x_2 x_3 x_4\} \\ &\quad - \mathrm{cum}(x_1; x_2)\mathrm{cum}(x_3; x_4) \\ &\quad - \mathrm{cum}(x_1; x_3)\mathrm{cum}(x_2; x_4) \\ &\quad - \mathrm{cum}(x_1; x_4)\mathrm{cum}(x_2; x_3). \end{aligned} \tag{4.37}$$

For independent random variables, these cumulants are guaranteed to be zero. For convenience, one can define the following second and fourth-order generalized autocorrelation functions for fourth-order stationary random processes,

$$C_x^{(2)}[k] \equiv \mathrm{cum}\left(x[n]; x[n+k]\right) \tag{4.38}$$

and

$$C_x^{(4)}[k_1, k_2, k_3] \equiv \mathrm{cum}\left(x[n]; x[n+k_1]; x[n+k_2]; x[n+k_3]\right). \tag{4.39}$$

Then, the whiteness requirement for the supex algorithm is that

$$C_x^{(2)}[k] = 0, \qquad k \neq 0 \tag{4.40}$$

and

$$C_x^{(4)}[k_1, k_2, k_3] = 0, \qquad (k_1, k_2, k_3) \neq (0, 0, 0). \tag{4.41}$$

For the case $\beta = 2$, if the initial condition $x[0]$ is drawn from a uniform distribution on the interval $[-1, 1]$, sequences generated by the tent map are uncorrelated and each $x[n]$ has the same distribution as $x[0]$ [5]. Thus, for these sequences

$$C_x^{(2)}[k] = \frac{1}{3}\delta[k] \tag{4.42}$$

as required by (4.40). Here, $\delta[k]$ is the usual unit sample function. However, the following argument shows that (4.41) is not satisfied. Without loss of generality, one can assume that $0 \leq k_1 \leq k_2 \leq k_3$. Then, from the definition of fourth-order cumulant,

$$\begin{aligned} C_x^{(4)}[k_1, k_2, k_3] &= E\left\{x[n]x[n+k_1]x[n+k_2]x[n+k_3]\right\} \\ &\quad - \frac{1}{9}\left(\delta[k_1]\delta[k_3 - k_2] + \delta[k_2]\delta[k_3] + \delta[k_3]\right). \end{aligned} \tag{4.43}$$

Also,

$$E\left\{x[n]x[n+k_1]x[n+k_2]x[n+k_3]\right\} = \int_{-1}^{1} xF^{(k_1)}(x)F^{(k_2)}(x)F^{(k_3)}(x)dx, \tag{4.44}$$

where $F^{(k)}(x)$ is the k-fold composition of the symmetric tent map. The function $F^{(k)}(x)$ is even for all $k > 0$, and $F^{(k)}(x) = x$ is odd for $k = 0$. There are three separate cases to consider in verifying (4.41).

1. $k_1 \neq 0, k_2 \neq 0, k_3 \neq 0$.

   The integrand in (4.44) is odd, so the integral is zero. By (4.43),

$$C_x^{(4)}[k_1, k_2, k_3] = 0. \tag{4.45}$$

2. $k_1 = 0, k_2 \neq 0, k_3 \neq 0$.

   The integrand in (4.44) is even, so the integral is nonzero. Then, when $k_2 \neq k_3$, it is guaranteed that

$$C_x^{(4)}[0, k_2, k_3] \neq 0. \qquad (4.46)$$

3. $k_1 = 0, k_2 = 0, k_3 \neq 0$.

   The integrand in (4.44) is odd, so the integral is zero. By (4.43),

$$C_x^{(4)}[0, 0, k_3] = 0. \qquad (4.47)$$

Therefore, the tent map sequence does not satisfy the fourth-order whiteness constraint in Case 2.

However, the supex algorithm depends on fourth-order whiteness only to the extent that $\left|C_x^{(4)}[k_1, k_2, k_3]\right| \ll \left|C_x^{(4)}[0, 0, 0]\right|$. Numerical calculations indicate that tent map signals satisfy this approximate whiteness constraint except in the case where $k_1 = 0$, $k_2 = 1$, and $k_3 = 2$. Table 4.1 shows normalized numerically calculated $\left|C_x^{(4)}[0, k_2, k_3]\right|$ with $k_2, k_3 \neq 0$. To calculate these cumulants, the integral in (4.44) was numerically evaluated using the trapezoid approximation, with a trapezoid width of 10^{-4}. The ratio of $\left|C_x^{(4)}[0, k_2, k_3]\right|$ to $\left|C_x^{(4)}[0, 0, 0]\right|$ is displayed in the table. The values for k_2 increase across rows of the table, and k_3 increases down columns. For example, the second entry in the third row is $\left|C_x^{(4)}[0, 2, 3]/C_x^{(4)}[0, 0, 0]\right|$.

The boldface numbers in Table 4.1 correspond to fourth-order generalized autocorrelations of the form $\left|C_x^{(4)}[0, 2k_2, 2k_3]\right|$. These cumulants are much smaller than $\left|C_x^{(4)}[0, 0, 0]\right|$. If $x_d[n]$ represents a downsampled by two version of $x[n]$, i.e., $x_d[n] = x[2n]$, then

$$\begin{aligned} \left|C_{x_d}^{(4)}[0, k_2, k_3]\right| &= \left|C_x^{(4)}[0, 2k_2, 2k_3]\right| \\ &\ll \left|C_x^{(4)}[0, 0, 0]\right| \\ &= \left|C_{x_d}^{(4)}[0, 0, 0]\right| \\ \Longrightarrow \left|C_{x_d}^{(4)}[0, k_2, k_3]\right| &\ll \left|C_{x_d}^{(4)}[0, 0, 0]\right|. \end{aligned} \qquad (4.48)$$

Table 4.1: Normalized Cumulants. The table gives numerically calculated normalized cumulants: $\left|\frac{C_x^{(4)}[0,k_2,k_3]}{C_x^{(4)}[0,0,0]}\right|$

		k_2					
		1	2	3	4	5	6
	1	0.1667					
	2	0.6250	**0.0417**				
k_3	3	0.1563	0.1563	0.0104			
	4	0.0391	**0.0391**	0.0391	**0.0026**		
	5	0.0098	0.0098	0.0098	0.0098	0.0007	
	6	0.0024	**0.0024**	0.0024	**0.0024**	0.0024	**0.0002**

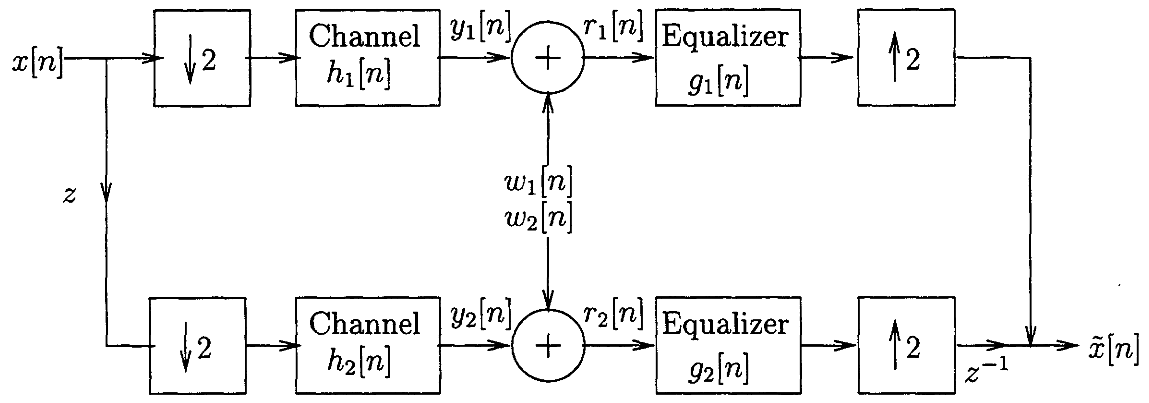

Figure 4-3: Downsampling and Supex for Equalization. Even and odd phases are processed over parallel, non-interfering channels.

Thus, downsampling tent map sequences by two approximately whitens their fourth-order spectra. Similar results apply for $x_d[n] = x[Mn]$, where $M \geq 2$. Therefore, the map

$$x[n] = F^{(M)}(x[n-1]), \tag{4.49}$$

generates fourth-order quasi-white sequences, so that the supex algorithm can be used in their deconvolution.

Supex Deconvolution of Whitened Tent Map Sequences

Figure 4-3 illustrates a system using the supex algorithm for deconvolution. A down-sampler whitens the tent map sequence $x[n]$, separating it into even and odd phases.

The even numbered samples pass through the upper channel and the odd numbered samples pass through the lower channel. The upper and lower channels represent two non-interfering or decoupled channels. For example, the even and odd phases could be sent at two different frequencies so that $h_1[n]$ and $h_2[n]$ could be the impulse responses of two channels with non-overlapping frequency responses, the sum of which is the frequency response of the overall channel $h[n]$. The supex algorithm separately examines the two received sequences, $r_1[n]$ and $r_2[n]$, to find the equalizer filter taps, $g_1[n]$ and $g_2[n]$. After equalizing the two phases separately, the system recombines the two phases to obtain the estimate of the full original sequence.

4.2 Performance

This section examines the performance of the block deconvolution algorithms presented in the previous section. The recursive dynamics matching algorithm is not discussed here, but it's performance in the context of decoding tent map encoded data is discussed in Section 4.3. Two performance measures, intersymbol interference and mean square error, are used to evaluate the algorithms. The effect of variables such as noise, initialization, signal length, and channel impulse response are considered.

4.2.1 Performance Measures

Two performance measures are used to evaluate the deconvolution algorithms, intersymbol interference (ISI) and mean square error (MSE). ISI measures how closely the equalizer removes the effect of the convolutional distortion caused by the channel. In particular, if $s[n] = h[n] * g[n]$ is the impulse response of the cascaded channel and equalizer,

$$\mathrm{ISI} = \frac{\sum_n s^2[n] - \max_n s^2[n]}{\max_n s^2[n]}. \tag{4.50}$$

Thus, ISI is an "interference to signal ratio" where the signal is an impulse of height $\max\{s[n]\}$ and the interference is the difference between this signal and $s[n]$. ISI is a useful performance measure when the chaotic sequence is used for probing the channel or training the equalizer since it measures how well the equalizer combats convolutional distortion.

The MSE measures how closely the estimate of the signal $\hat{x}[n]$ matches the actual signal $x[n]$. Specifically,

$$\text{MSE} = \frac{1}{N}\sum_n \left(\hat{x}[n] - x[n]\right)^2, \tag{4.51}$$

where $\hat{x}[n]$ is the result of processing the equalizer output $\tilde{x}[n]$ with the ML estimator for tent map sequences in AWGN. (See Figure 4-1.) MSE is a useful performance measure when the chaotic sequence is the signal of interest which one wishes to recover.

4.2.2 Test Channels

The following channels are used to test the algorithms. The constants, K, in each case are chosen so that the channel has unit energy.

- The FIR channel

$$H_{\text{FIR}}(z) = K_{\text{FIR}}(0.4 + z^{-1} - 0.7z^{-2} + 0.6z^{-3} + 0.3z^{-4} - 0.4z^{-5} + 0.1z^{-6}) \tag{4.52}$$

- The all-pole channel

$$H_{\text{aple}}(z) = \frac{K_{\text{aple}}}{1 + 0.78z^{-1} + 0.58z^{-2} - 0.04z^{-3} - 0.11z^{-4} - 0.0882z^{-5}} \tag{4.53}$$

- The all-pass channel

$$H_{\text{apss}} = \frac{1 + 0.78z^{-1} + 0.58z^{-2} - 0.04z^{-3} - 0.11z^{-4} - 0.0882z^{-5}}{-0.0882 - 0.11z^{-1} - 0.04z^{-2} + 0.58z^{-3} + 0.78z^{-4} + z^{-5}} \tag{4.54}$$

The FIR channel is the test channel, normalized to unit energy, used by Shalvi and Weinstein [6]. The all-pole and all-pass channels are the same test channels, normalized to unit energy, used by Isabelle [4]. In all experiments testing the supex algorithm, the same channel distorts the even and odd phases, i.e., $h_1[n] = h_2[n]$ in Figure 4-3.

4.2.3 Convergence Curves

Figures 4-4, 4-5, and 4-6 illustrate the convergence behavior of the three algorithms for the three test channels described above. Each of the curves represents average ISI and MSE at each iteration. White Gaussian noise at a SNR of 20 dB was added to filtered tent map sequences of length 1024. For the supex algorithm, there were actually two sequences of length 1024, the even and odd phases of a length 2048 tent map sequence. Each of the three algorithms was tested on the noisy, convolutionally distorted tent map sequences, and the ISI and MSE for 50 trials were measured. The initial guess for the equalizer was a 16-tap filter, z^{-3}. Any trial with MSE < 0.1 was considered a success. Only the successful trials were kept for the algorithm with the least successes, and the same number were kept for the other two algorithms. For example, if the algorithm with the least number of successess was successful 45 out of the 50 trials, the top 45 trials in terms of lowest MSE were kept for each of the algorithms. These trials were averaged together, and the results are plotted in Figures 4-4, 4-5, and 4-6.

Because the dynamics matching and alternating projections algorithms make explicit use of the deterministic structure of the tent map sequences, one might expect that they converge more quickly than the supex algorithm. Indeed, Figures 4-4 and 4-6 support this hypothesis in the cases of the FIR and all-pass channels. However, in the case of the all-pole channel (Figure 4-5), the experimental data does not support the hypothesis. The reason for this counter-intuitive result is not known at this time.

Finally, the three algorithms seem to perform most poorly on the all-pass channel. It is unclear whether this phenomenon is related to the fact that the all-pass channel has a flat magnitude spectrum since the three algorithms are all derived and implemented in the time domain.

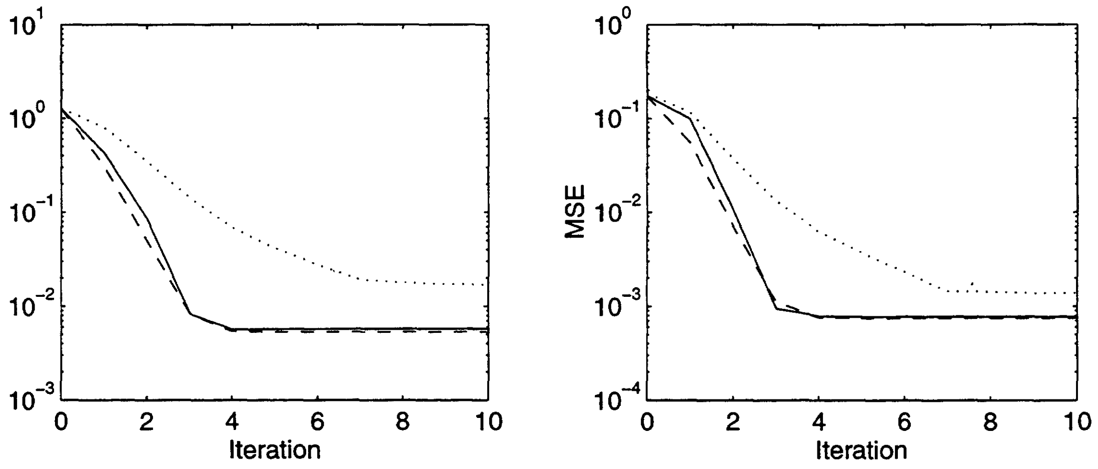

Figure 4-4: Deconvolution Performance on the FIR Channel. The solid, dashed, and dotted curves represent the performance of the dynamics matching, alternating projections, and supex algorithms, respectively. SNR = 20 dB. $N = 1024$.

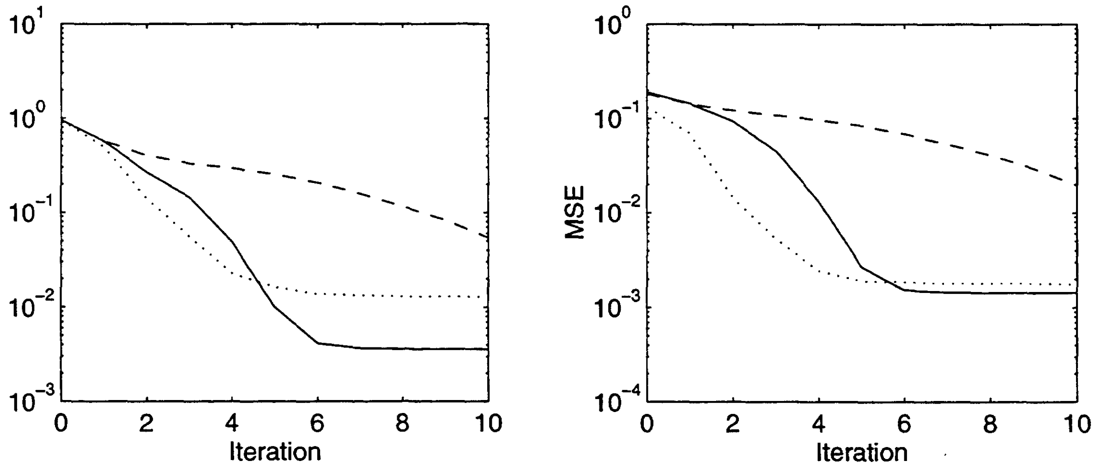

Figure 4-5: Deconvolution Performance on the All-Pole Channel. The solid, dashed, and dotted curves represent the performance of the dynamics matching, alternating projections, and supex algorithms, respectively. SNR = 20 dB. $N = 1024$.

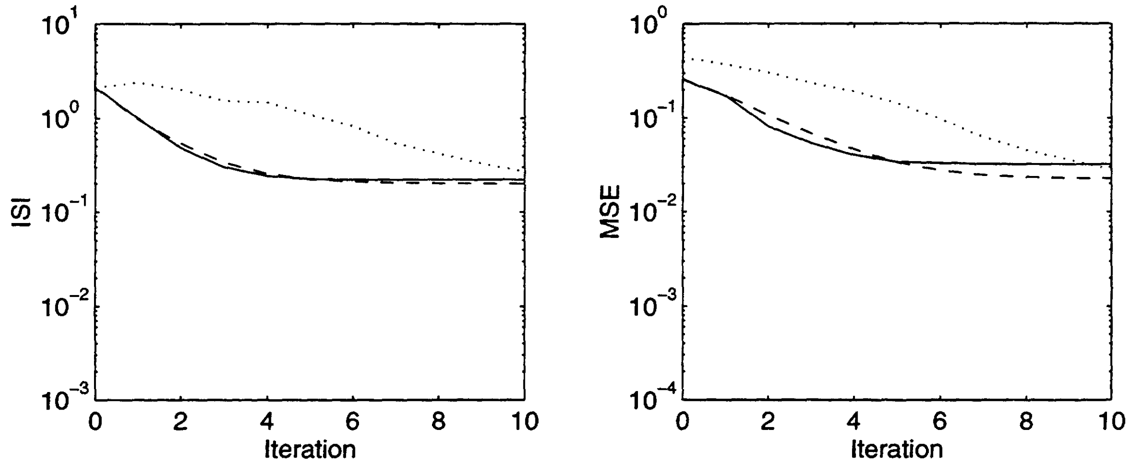

Figure 4-6: Deconvolution Performance on the All-Pass Channel. The solid, dashed, and dotted curves represent the performance of the dynamics matching, alternating projections, and supex algorithms, respectively. SNR = 20 dB. $N = 1024$.

4.2.4 Sequence Length Effects

Figures 4-7, 4-8, and 4-9 demonstrate the dependence of algorithm performance on sequence length (N). As one might expect, longer sequences lead to lower ISI and MSE, although there appears to be a lower limit on these performance measures in the case of the dynamics matching and alternating projections algorithms.

Once again, the three algorithms were tested on 50 noisy, convolutionally distorted tent map sequences at SNRs of 10, 20, and 30 dB, and sequence lengths of $100, 200, \cdots, 1000$. The same initialization as before (16-tap, z^{-3}) was used, and the channel was the FIR filter used above. In this experiment, however, no trials were discarded before averaging the ISI and MSE. The average ISI and MSE after 10 iterations is shown in Figures 4-7, 4-8, and 4-9 for the dynamics matching, alternating projections, and supex algorithms, respectively.

The figures show that ISI and MSE both decrease with longer N. However, in the case of the dynamics matching and alternating projections algorithms, the ISI and MSE curves seem to become flat after reaching a lower limit. This phenomenon is not very surprising in light of Chapter 2, which calculated a lower threshold for MSE in the case where AWGN, but not convolutional distortion, was present.

Curiously, Figure 4-8 seems to imply that for some sequence lengths ($200 \leq N \leq$

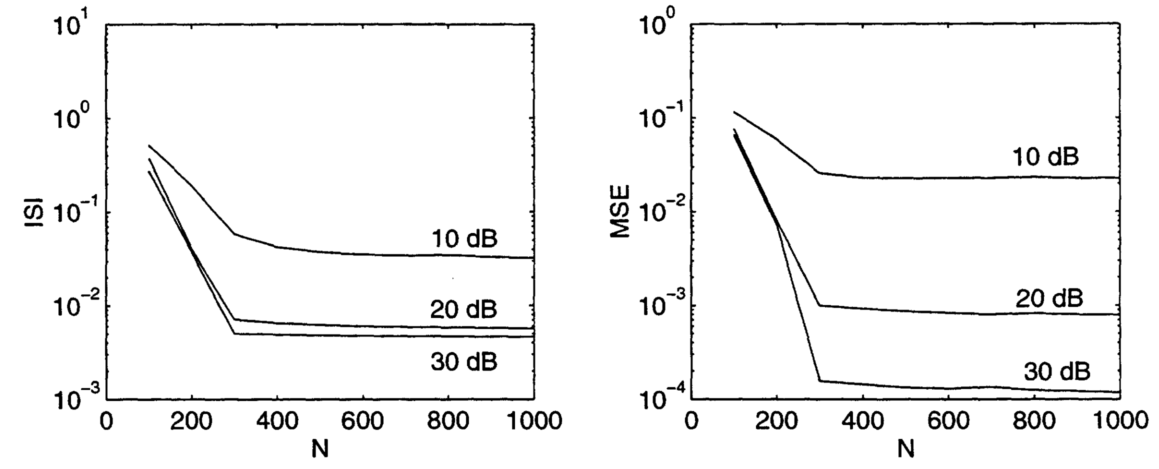

Figure 4-7: Effect of Sequence Length (N) on Dynamics Matching Algorithm. The channel is the FIR channel.

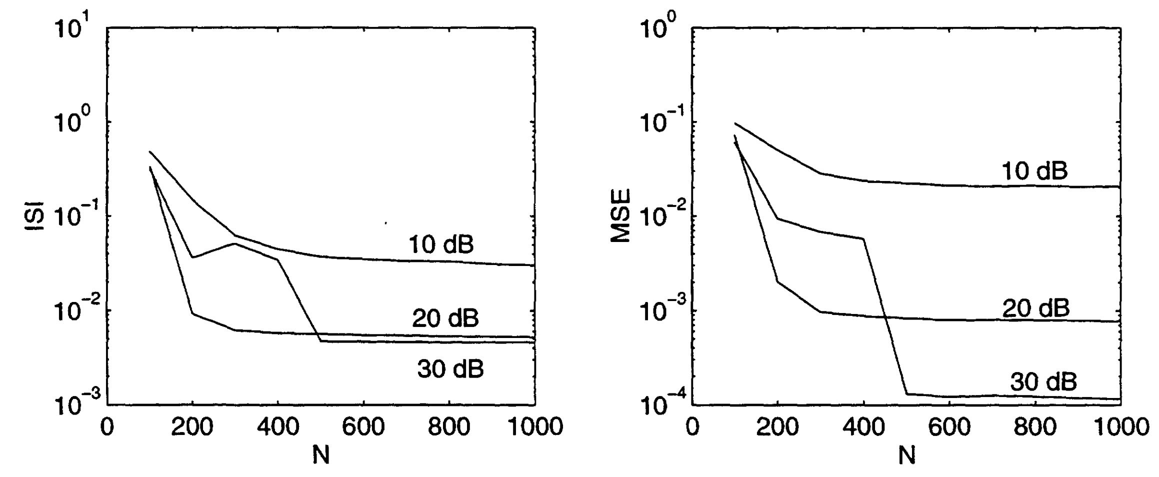

Figure 4-8: Effect of Sequence Length (N) on Alternating Projections Algorithm. The channel is the FIR channel.

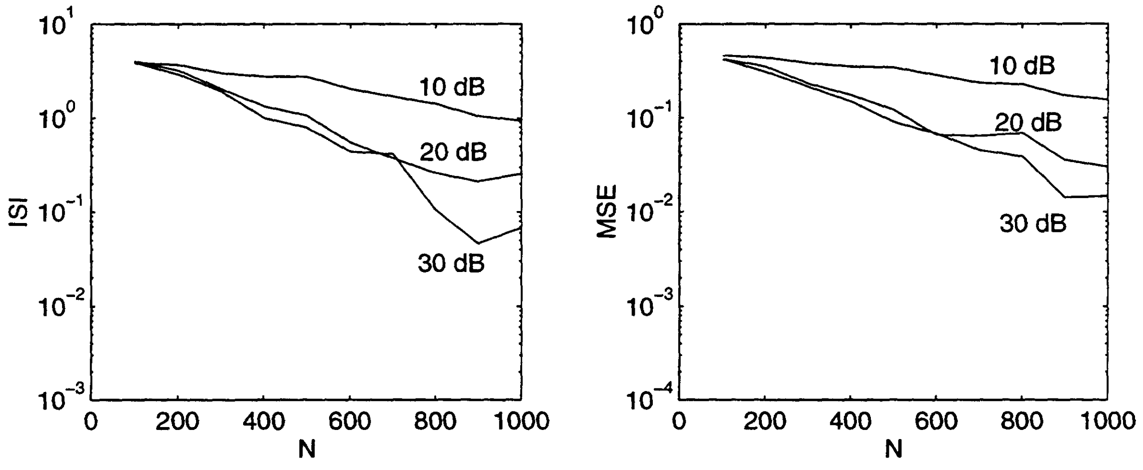

Figure 4-9: Effect of Sequence Length (N) on Supex Algorithm. The channel is the FIR Channel.

400) the alternating projections algorithm performs better when the SNR is 20 dB than when the SNR is 30 dB. The reason for this anamolous result is not known at this time.

4.2.5 Behavior in Noise

This section relates the performance of the three algorithms to the SNR. The experiment was similar to the one in Subsection 4.2.4 above, except that SNRs of $0, 5, \cdots, 40$ dB and sequence lengths of 256 and 512 were used. The other parameters were the same as in Subsection 4.2.4. Figures 4-10, 4-11, and 4-12 show the results of the experiment.

Although higher SNRs usually correspond to lower ISI and MSE, the curve for N=256 in Figure 4-10 again demonstrates an anamoly where the performance of the dynamics matching algorithm is better with an SNR of 25 dB than with an SNR between 30 and 40 dB. Recall, Figure 4-8 in Subsection 4.2.4 demonstrated this same anamoly for the alternating projections algorithm. The curves in Subsection 4.2.4 and in this subsection suggest that this anamolous effect occurs only for short sequence lengths, i.e., sequences with $N < 400$.

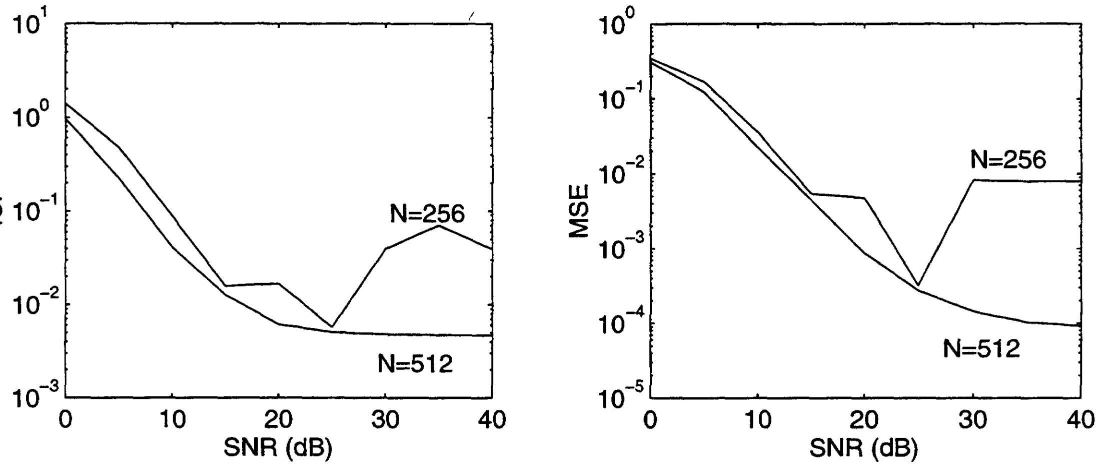

Figure 4-10: Effect of Noise on Dynamics Matching Algorithm. The channel is the FIR Channel.

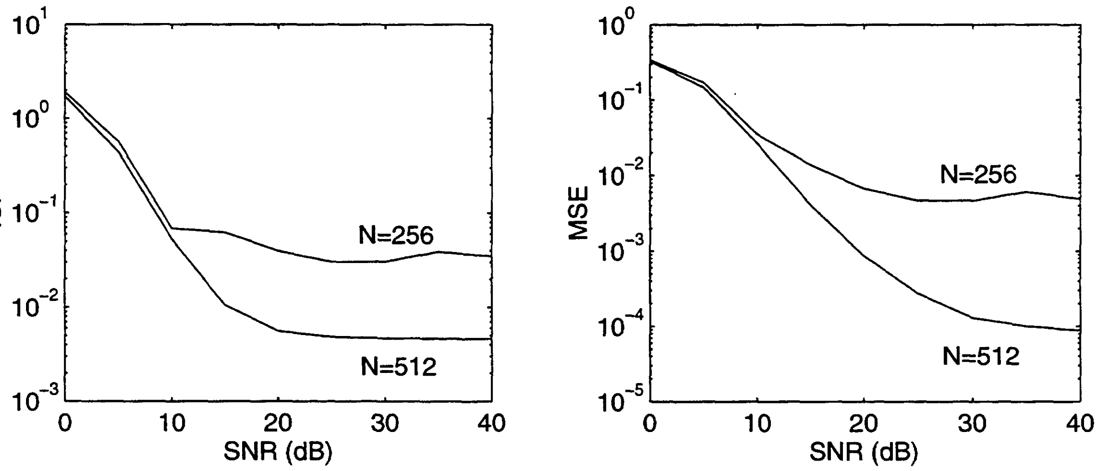

Figure 4-11: Effect of Noise on Alternating Projections Algorithm. The channel is the FIR Channel.

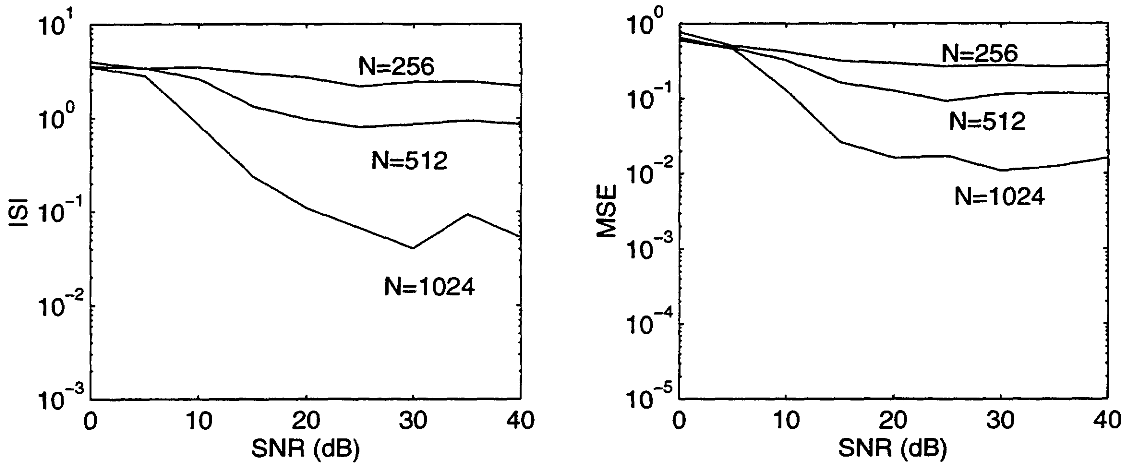

Figure 4-12: Effect of Noise on Supex Algorithm. The channel is the FIR Channel.

4.2.6 Initialization and Convergence

An important issue in considering the performance of an iterative algorithm is the question of whether the algorithm will converge under a given initialization and whether it converges to the correct solution. This subsection examines the convergence of the three deconvolution algorithms and describes the effect of initialization on each.

Dynamics Matching and Local Minima

As stated in Subsection 4.1.1, the dynamics matching algorithm converges to the global minimum of $J_{\hat{\mathbf{s}}}(\mathbf{g})$ rather than the global minimum of the ultimate objective function $J(\mathbf{g})$. Since $\mathbf{s}(\mathbf{g}) = \mathbf{s}(\mathbf{g}+\delta)$ if δ is chosen small enough, the global minimum of $J_{\hat{\mathbf{s}}}(\mathbf{g})$ is actually a local minimum of $J(\mathbf{g})$. However, this objective function has several local minima, in general. Figure 4-13 illustrates this point.

Figure 4-13 is a plot of $J(\mathbf{g})$ when the equalizer is a 2-tap FIR filter, the channel has only a single pole at $z = -0.5$, and there is no noise. Note that the globally minimum point $\hat{\mathbf{g}} = [1\ 0.5]^T$ is not the only local minimum point. Thus, the dynamics matching algorithm is not guaranteed to converge to the global minimum of $J(\mathbf{g})$.

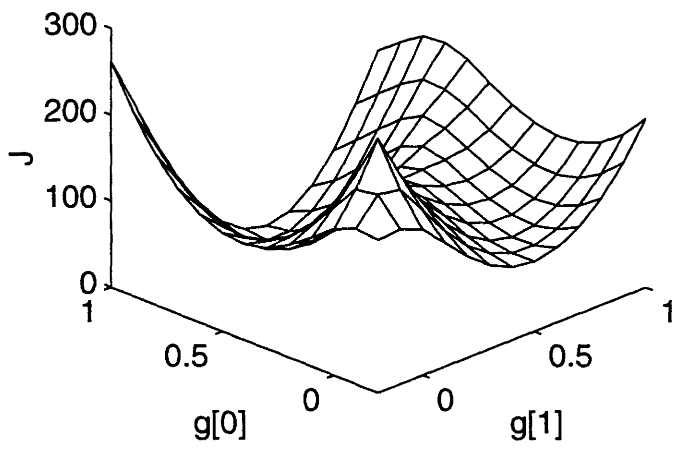

Figure 4-13: The Dynamics Matching Objective Function $J(\mathbf{g})$ for 2-tap Equalizer. The channel has only a single pole at $z = -0.5$, no noise.

Alternating Projections and Non-convexity

The iterative minimization in the alternating projections algorithm converges to a global minimum if both the equalizer output set ($\mathcal{O}$) and the set of tent map sequences ($\mathcal{T}$) are convex. However, $\mathcal{T}$ is not convex. One can, however, show that the algorithm converges monotonically to a local minimum.

Claim 3 *The alternating projections algorithm converges monotonically to a local minimum of* $||\mathbf{x} - \mathbf{R}\mathbf{g}||^2$.

Proof. Step 2 of the alternating projections algorithm implies that

$$||\hat{\mathbf{x}}^{(i)} - \mathbf{R}\hat{\mathbf{g}}^{(i-1)}||^2 \leq ||\mathbf{x} - \mathbf{R}\hat{\mathbf{g}}^{(i-1)}||^2, \qquad \forall \mathbf{x} \in \mathcal{T}. \tag{4.55}$$

Step 3 implies that

$$||\hat{\mathbf{x}}^{(i)} - \mathbf{R}\hat{\mathbf{g}}^{(i)}||^2 \leq ||\hat{\mathbf{x}}^{(i)} - \mathbf{R}\mathbf{g}||^2, \qquad \forall \mathbf{g}. \tag{4.56}$$

Thus,

$$\begin{aligned} ||\hat{\mathbf{x}}^{(i-1)} - \mathbf{R}\hat{\mathbf{g}}^{(i-1)}||^2 &\geq ||\hat{\mathbf{x}}^{(i)} - \mathbf{R}\hat{\mathbf{g}}^{(i-1)}||^2 \quad \text{by (4.55)} \\ &\geq ||\hat{\mathbf{x}}^{(i)} - \mathbf{R}\hat{\mathbf{g}}^{(i)}||^2 \quad \text{by (4.56)}. \end{aligned} \tag{4.57}$$

Hence, $||\hat{\mathbf{x}} - \mathbf{R}\hat{\mathbf{g}}||^2$ is monotonically non-increasing with each iteration.

Since $|| \cdot ||^2 \geq 0$, eventually $||\hat{\mathbf{x}}^{(i)} - \mathbf{R}\hat{\mathbf{g}}^{(i-1)}||^2 = ||\hat{\mathbf{x}}^{(i)} - \mathbf{R}\hat{\mathbf{g}}^{(i)}||^2$. Then, since $\hat{\mathbf{g}}^{(i)}$ uniquely minimizes $||\hat{\mathbf{x}}^{(i)} - \mathbf{R}\mathbf{g}||^2$, one can conclude that $\hat{\mathbf{g}}^{(i)} = \hat{\mathbf{g}}^{(i-1)}$. Therefore, the algorithm converges to some $(\hat{\mathbf{x}}, \hat{\mathbf{g}})$ where

$$||\hat{\mathbf{x}} - \mathbf{R}\hat{\mathbf{g}}||^2 \leq ||\mathbf{x} - \mathbf{R}\hat{\mathbf{g}}||^2, \qquad \forall \mathbf{x} \in \mathcal{T}. \tag{4.58}$$

and

$$||\hat{\mathbf{x}} - \mathbf{R}\hat{\mathbf{g}}||^2 \leq ||\hat{\mathbf{x}} - \mathbf{R}\mathbf{g}||^2, \qquad \forall \mathbf{g}. \tag{4.59}$$

Therefore, the gradient of $||\mathbf{x} - \mathbf{R}\mathbf{g}||^2$ is zero at $(\hat{\mathbf{x}}, \hat{\mathbf{g}})$, i.e., the algorithm converges monotonically to a local minimum of $||\mathbf{x} - \mathbf{R}\mathbf{g}||^2$.
□

Supex

When the channel input is a sequence of random variables satisfying the second and fourth order whiteness constraints (4.40) and (4.41), the impulse response of the cascaded channel and equalizer approaches a delay with each iteration of the supex algorithm, regardless of the initialization [6]. Hence, the ISI should decrease with each iteration. However, when the supex algorithm was used on tent map sequences, there were many instances where the ISI did not decrease with each iteration, and the supex algorithm did not yield a "good" equalizer. Apparently, the approximate whitening performed by the downsampler is not sufficient in some cases.

4.3 Coding for the ISI Channel

Slight modifications to the deconvolution algorithms in this chapter yield decoding algorithms for tent map encoded data which has been transmitted over an unknown ISI channel with AWGN. In particular, suppose $x_0[0], x_0[1], \cdots, x_0[L-1]$ is a sequence of independent random variables uniformly distributed on the interval $[-1, 1]$ representing the source data. The encoded sequence $x[n]$ is a concatenation of L length-N sequences, each of which corresponds to one of the $x_0[k]$. Formally,

$$x[n] = \sum_{k=0}^{L-1} \sum_{l=kN}^{(k+1)N-1} F^{(n-kN)}(x_0[k])\delta[n-l], \tag{4.60}$$

where $F^{(n)}(\cdot)$ is the n-fold composition of the symmetric tent map. The decoder for such sequences consists of the equalizer-noise removal structure in Figure 4-1 followed by a sampler which samples at integer multiples of N. These are the samples of $\hat{x}[n]$ that correspond to the source data $x_0[k]$, and any delay in the cascaded channel-equalizer system is assumed to be known so that one knows the points at which to sample.

4.3.1 Decoding Algorithms

The algorithms for choosing the equalizer taps when the signals have the form (4.60) are very similar to the deconvolution algorithms from earlier parts in this chapter. The modifications which are needed are given below.

Dynamics Matching Decoding

Concatenated tent map sequences of the form given by (4.60) satisfy (4.1) for all $n \neq kN$, where k is an integer. Thus, the only required modification to the block dynamics matching algorithm presented in Subsection 4.1.1 is to remove the rows of the matrix $\mathbf{R_s}$ which correspond to $n = kN$. In the recursive implementation, the recursion (4.28) and (4.29) occurs only when the new data does not correspond to these removed rows. The rest of the algorithms remain the same.

Alternating Projections Decoding

To use the alternating projections algorithm for concatenated tent map sequences, one needs only to replace the set of tent map sequences ($\mathcal{T}$) with the set of concatenated tent map sequences ($\mathcal{T}_{\text{cat}}$), the set of all sequences of the form (4.60). Then, to find the $\hat{\mathbf{x}}^{(i)}$ in $\mathcal{T}_{\text{cat}}$ which minimizes $\|\mathbf{x} - \mathbf{R}\hat{\mathbf{g}}^{(i-1)}\|^2$ (Step 2 in the alternating projections algorithm), one must parse $\mathbf{R}\hat{\mathbf{g}}^{(i-1)}$ into L sequences of length N and filter and smooth the individual parsed sequences with the ML-AWGN estimator from Chapter 2.

Supex Decoding

The supex algorithm works on any sequence satisfying the second and fourth order cumulant whiteness constraints (4.40) and (4.41). Downsampled concatenated tent map sequences approximately satisfy the whiteness constraints at least as well as single tent map sequences. Thus, the supex algorithm with downsampling discussed earlier requires no modification to work on concatenated tent map sequences.

4.3.2 Simulation Results

The section presents the experimental results when the tent map code was used in the transmission of a uniform source over an ISI channel with AWGN noise.

Block Decoding

The three block decoding algorithms discussed above were tested on a source data sequence of 1024 independent, identically distributed (IID) random variables uniformly distributed on the interval $[-1, 1]$. Each of these variables was mapped onto a tent map sequence of length N, and L such sequences were concatenated together to form a block of length $NL = 1024$. (For the supex algorithm, L was chosen to make $NL = 2048$ so that each phase had length 1024 after downsampling.) Each algorithm was run on all of the blocks, and the average ISI across all blocks was calculated, as was the mean square error distortion (D) in estimating the 1024 source data variables. The performance of the supex algorithm on uncoded source sequences serves as an arbitrary but useful baseline. The improvement in ISI and distortion over this baseline is shown in Figures 4-14 and 4-15, respectively. The SNR and N were chosen to correspond to points in the power-bandwidth plane where tent map coding beats repetition coding for transmission over the AWGN (no ISI) channel. (See Figure 3-4.)

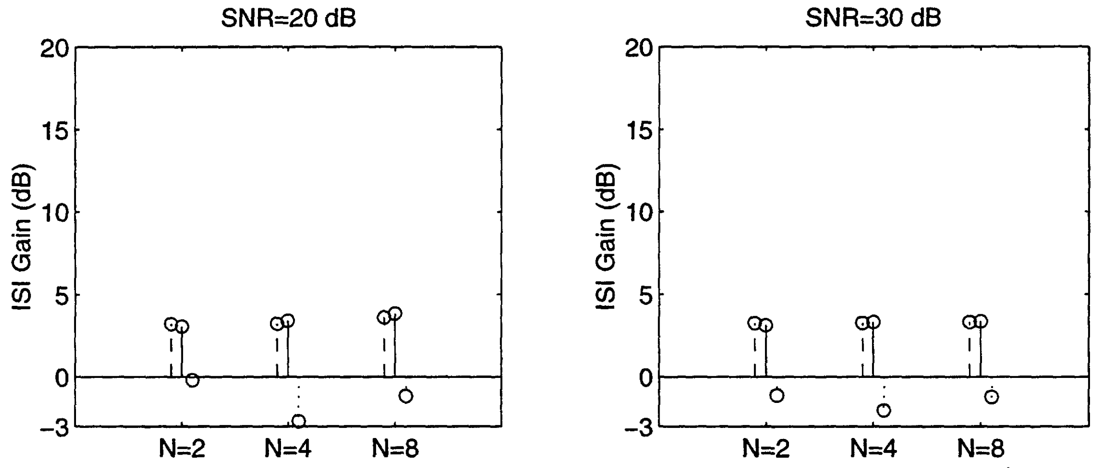

Figure 4-14: Improvement over baseline in ISI using dynamics matching (dashed), alternating projections (solid), and supex (dotted) algorithms.

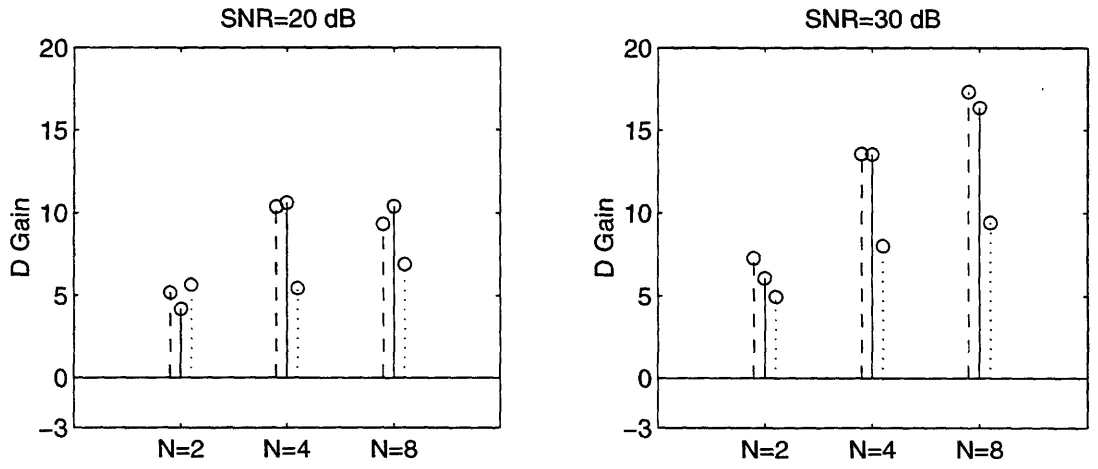

Figure 4-15: Improvement over baseline in Distortion using dynamics matching (dashed), alternating projections (solid), and supex (dotted) algorithms.

If one quickly glances only at Figure 4-14, one might conclude that tent map coding offers only moderate improvement in residual ISI over the baseline of using the supex algorithm on an uncoded sequence. Indeed, one would expect zero ISI gain for the case $N = 2$ when supex decoding is used since the even and odd phases are in fact IID sequences. However, if one examines Figure 4-15, one will notice that tent map coding offers substantial improvement in minimizing distortion over the baseline case, especially if dynamics matching or alternating projections decoding is used. Thus, if one interprets residual ISI as a kind of noise, one could say that just as tent map coding can reduce distortion caused by AWGN, it can reduce "residual ISI noise" as well. Furthermore, that the gain in distortion reduction increases with SNR, even at fairly high SNR, suggests that the results demonstrating the superiority of tent map coding to M-ary coding at high power in the AWGN channel case may also apply to the ISI channel case.

Recursive Dynamics Matching Decoding

To test the performance of the recursive dynamics matching algorithm on concatenated tent map sequences, an IID sequence of 1024 random variables uniformly distributed on the interval $[-1, 1]$ was encoded onto 1024 concatenated tent map sequences of length 4. The concatenated sequences were filtered by the FIR channel used in previous sections and noise at an SNR of 20 dB was added. Recall from Chapter 3 that for $N = 4$ and SNR $= 20$ dB the tent map code beats any linear modulation code and any code with a binary channel alphabet.

Note that the recursion defined by Equations (4.28) and (4.29) can only begin after $\mathbf{R}_{\hat{\mathbf{s}}}$ has more rows than columns. (This condition is necessary for ${\mathbf{R}_{\hat{\mathbf{s}}}}^T\mathbf{R}_{\hat{\mathbf{s}}}$ to be invertible.) Thus, to initialize the algorithm, the (non-recursive) dynamics matching algorithm was run on a distorted tent map sequence of length $2M$, where M was the number of equalizer taps. The resulting $\mathbf{R}_{\hat{\mathbf{s}}_{2M-1}}$ and $\hat{\mathbf{g}}_{2M-1}$ were used to initialize the recursive algorithm.

Figure 4-16 shows the performance, measured in terms of ISI, of the recursive dynamics matching algorithm on the concatenated tent map sequence. Note the

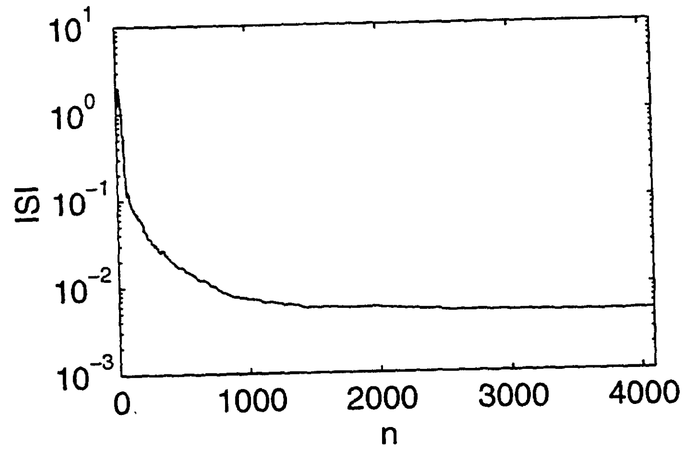

Figure 4-16: Performance of the Recursive Dynamics Matching Algorithm. $N = 4$, SNR = 20 dB.

improvement in ISI over time. For comparison, in the experiment used to generate Figure 4-14 when $N = 4$ and SNR = 20 dB, the ISI for the block dynamics matching algorithm is 0.0060. The ISI after 1024 samples ($n = 1023$) for the recursive algorithm is 0.0073.

Chapter 5

The Multiple Access Channel

In some communications applications, such as in wireless communications, many users send messages over the same channel. Such a channel is called a multiple access channel, and one may view any particular user's message as *multiple access interference* which hinders the recovery of some other user's message. The receiver for these channels observes the superposition of all the users' messages and must be able to separate them to obtain the message or messages of interest. Therefore, if one wishes to use chaotic signals in a communications context, it is helpful to have algorithms for their estimation in the presence of multiple access interference. Although the work in this chapter has been motivated by the communications applications mentioned above, it should be emphasized that signal separation algorithms may be useful in a variety of other contexts as well. For example, in a remote sensing context, the superimposed signals may represent signals from multiple objects or targets.

This chapter considers the separation of superimposed tent map sequences. The maximum likelihood estimator for the joint estimation of tent map sequences that have been transmitted synchronously over a multiple access channel with additive white Gaussian noise is developed. Such an estimator could be a decoder for tent map encoded data, for example.

5.1 Problem Formulation

The problem considered is the following: one observes the superposition of m tent map sequences of length N in AWGN,

$$\begin{aligned} y[k] &= \sum_{i=1}^{m} x_i[k] + w[k], \qquad k = 0, \cdots, N-1 \\ \mathbf{K}_{ww} &= \operatorname{diag}\left(\sigma^2_{w[0]}, \cdots, \sigma^2_{w[N-1]}\right) \end{aligned} \tag{5.1}$$

and wishes to obtain estimates $\hat{x}_i[k]$ that are optimal under the maximum likelihood criterion. The reader should note that this problem considers only perfectly synchronized signals with uniform amplitudes. The asynchronous case with multiple amplitudes is reserved for future work.

For the same reasons as in Chapter 2, one may decompose the ML estimator into a filtering stage followed by a smoothing stage, and in the filtering stage, one may perform the maximization of the likelihood function over the signs separately from the maximization over the $x_i[n]$. Specifically,

$$\{\hat{x}_i[n|n]\}_i = \arg\max_{\{x_i[n]\}_i} \ln p\left(\mathbf{y}_n | \{\hat{\mathbf{s}}_i[0:n-1]\}_i, x[n]\right), \tag{5.2}$$

where $\{\cdot\}_i$ denotes a set whose elements are indexed by i, e.g.,

$$\begin{aligned} \{x_i[n]\}_i &= \{x_1[n], x_2[n], \cdots, x_m[n]\} \\ \{\hat{\mathbf{s}}_i[0:n-1]\}_i &= \{\hat{\mathbf{s}}_1[0:n-1], \cdots, \hat{\mathbf{s}}_m[0:n-1]\}, \end{aligned}$$

and where

$$\mathbf{s}_i[k:n] = [s_i[k] \ \cdots \ s_i[n]]^T \tag{5.3}$$

$$\mathbf{y}_n = [y[0] \ \cdots \ y[n]]^T. \tag{5.4}$$

Then, the ML filtering problem reduces to the following minimization,

$$\{\hat{x}_i[n|n]\}_i = \arg\min_{\{x_i[n]\}_i} \sum_{k=0}^{n} \frac{1}{\sigma^2_{w[k]}} \left[y[k] - \sum_{i=0}^{m} F^{(k-n)}_{\hat{\mathbf{s}}_i[k:n-1]}(x_i[n]) \right]^2 . \tag{5.5}$$

As in the single-user case (Chapter 2), once this solution is obtained, one can proceed forward through the data since

$$\hat{s}_i[n] = \operatorname{sgn} \hat{x}_i[n|n]. \tag{5.6}$$

After filtering, one can smooth the estimates via

$$\hat{x}_i[n|N-1] = F^{(n-(N-1))}_{\hat{\mathbf{s}}_i[n:N-2]}(\hat{x}_i[N-1|N-1]). \tag{5.7}$$

5.2 Estimator Derivation

The minimization of (5.5) is facilitated by introducing the following auxiliary function:

$$\tilde{F}_{\{s_1,\cdots,s_m\}}(y) \equiv (s_1 + s_2 + \cdots + s_m)(\beta - 1) - \beta y, \qquad s_i = \pm 1. \tag{5.8}$$

One can also denote the k-fold iteration of this function as

$$\tilde{F}^{(k)}_{\{s_{1i}\}_i,\{s_{2i}\}_i,\cdots,\{s_{ki}\}_i}(y) \equiv \tilde{F}_{\{s_{ki}\}_i} \circ \tilde{F}_{\{s_{k-1,i}\}_i} \circ \cdots \circ \tilde{F}_{\{s_{1i}\}_i}(y) \tag{5.9}$$

This auxiliary function is related to the tent map through the following two identities:

$$\begin{aligned} \tilde{F}_{\{s_i\}_i}\left(\sum_i^m x_i\right) &= \sum_{i=1}^{m} [s_i(\beta-1) - \beta x_i] \\ &= \sum_{i=1}^{m} s_i\left[(\beta-1) - \beta s_i x_i\right] \\ &= \sum_{i=1}^{m} s_i F_{s_i}(x_i) \end{aligned} \tag{5.10}$$

and

$$\begin{aligned} a-\sum_{i=0}^{m} s_i' F_{s_i}^{(-1)}(b_i) &= a-\sum_{i=0}^{m} s_i' s_i \frac{\beta-1-b_i}{\beta} \\ &= -\frac{1}{\beta}\left[\left(\sum_{i=0}^{m} s_i' s_i\right)(\beta-1)-\beta a-\sum_{i=0}^{m} s_i' s_i b_i\right] \\ &= -\frac{1}{\beta}\left[\tilde{F}_{\{s_i' s_i\}_i}(a)-\sum_{i=0}^{m}(s_i' s_i) b_i\right]. \end{aligned} \tag{5.11}$$

Thus, the function $\tilde{F}_{\{s_i\}_i}(\cdot)$ connects the sum of tent map sequences at a given time to a combination of their sums and differences at some later time. In particular, by repeatedly applying the identity (5.11), one obtains

$$\begin{aligned} y[k]-\sum_{i=0}^{m} F_{\hat{\mathbf{s}}_i[k:n-1]}^{(k-n)}(x_i[n]) &= \left(-\frac{1}{\beta}\right)^{n-k}\left[\tilde{F}^{(n-k)}_{\{\hat{s}_i[k]\}_i,\left\{\prod_{l=k}^{k+1}\hat{s}_i[l]\right\}_i,\ldots,\left\{\prod_{l=k}^{n-1}\hat{s}_i[l]\right\}_i}(y[k])\right. \\ &\quad \left.-\sum_{i=1}^{m}\left(\prod_{l=k}^{n-1}\hat{s}_i[l]\right)x_i[n]\right]. \end{aligned} \tag{5.12}$$

Therefore, the minimization (5.5) can be written as

$$\begin{aligned} \{\hat{x}_i[n|n]\}_i &= \underset{\{x_i[n]\}_i}{\arg\min}\sum_{k=0}^{n}\left(\frac{1}{\beta^{2(n-k)}\sigma_{w[k]}^2}\right)\left[\tilde{F}^{(n-k)}_{\{\hat{s}_i[k]\}_i,\left\{\prod_{l=k}^{k+1}\hat{s}_i[l]\right\}_i,\ldots,\left\{\prod_{l=k}^{n-1}\hat{s}_i[l]\right\}_i}(y[k])\right. \\ &\quad \left.-\sum_{i=1}^{m}\left(\prod_{l=k}^{n-1}\hat{s}_i[l]\right)x_i[n]\right]^2, \end{aligned} \tag{5.13}$$

which is equivalent to minimizing $||\mathbf{e}||^2=\mathbf{e}^T\mathbf{S}_n\mathbf{e}$ in

$$\underbrace{\begin{bmatrix} \tilde{F}^{(n)}_{\{\hat{s}_i[0]\}_i,\left\{\prod_{l=0}^{1}\hat{s}_i[l]\right\}_i,\ldots,\left\{\prod_{l=0}^{n-1}\hat{s}_i[l]\right\}_i}(y[0]) \\ \tilde{F}^{(n-1)}_{\{\hat{s}_i[1]\}_i,\left\{\prod_{l=1}^{2}\hat{s}_i[l]\right\}_i,\ldots,\left\{\prod_{l=1}^{n-1}\hat{s}_i[l]\right\}_i}(y[1]) \\ \vdots \\ \tilde{F}_{\{\hat{s}_i[n-1]\}_i}(y[n-1]) \\ y[n] \end{bmatrix}}_{\tilde{\mathbf{y}}} = \underbrace{\begin{bmatrix} \prod_{l=0}^{n-1}\hat{s}_1[l] & \cdots & \prod_{l=0}^{n-1}\hat{s}_m[l] \\ \prod_{l=1}^{n-1}\hat{s}_1[l] & \cdots & \prod_{l=1}^{n-1}\hat{s}_m[l] \\ \vdots & \vdots & \vdots \\ \hat{s}_1[n-1] & \cdots & \hat{s}_m[n-1] \\ 1 & \cdots & 1 \end{bmatrix}}_{\tilde{\mathbf{A}}} \mathbf{x}[n|n]+\mathbf{e} \tag{5.14}$$

with $\mathbf{S}_n$ the same as in Chapter 2, i.e.,

$$\mathbf{S}_n = \mathrm{diag}(s_{0,n}, s_{1,n}, \cdots, s_{n,n}), \tag{5.15}$$

where

$$s_{i,n} = \frac{1}{\beta^{2(n-i)}\sigma^2_{w[i]}}. \tag{5.16}$$

Therefore, the ML estimates $\{\hat{x}_i[n|n]\}_i$ are the elements of the vector

$$\hat{\mathbf{x}}[n|n] = (\tilde{\mathbf{A}}^T\mathbf{S}_n\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{A}}^T\mathbf{S}_n\tilde{\mathbf{y}}. \tag{5.17}$$

This solution has a similar form to the single user solution (2.26). In the single user case, one uses $F_{\hat{\mathbf{s}}}(\cdot)$ to transform noisy observations of $x[k]$ into estimates of $x[n]$ which are linearly combined to form $\hat{x}[n|n]$. In the multiple user case, one uses $\tilde{F}_{\{\hat{\mathbf{s}}_i\}_i}(\cdot)$ to transform noisy observations of the sums of $\{x_i[k]\}_i$ into estimates of the sums and differences of $\{x_i[n]\}_i$ which are combined to form $\{\hat{x}_i[n|n]\}_i$.

One complication in the multiuser case that does not occur in the single user case, though, is that $(\tilde{\mathbf{A}}^T\mathbf{S}_n\tilde{\mathbf{A}})^{-1}$ exists only when $\tilde{\mathbf{A}}$ has linearly independent columns. In the single user case $\tilde{\mathbf{A}}$ is a single column vector $\mathbf{1}$, so this condition is always met. In the $m = 2$ case the columns of $\tilde{\mathbf{A}}$ are independent as long as $\hat{\mathbf{s}}_1[0:n-1] \neq \hat{\mathbf{s}}_2[0:n-1]$. One way to guarantee this condition in a tent map coding scenario, for example, is to set $s_1[0] \neq s_2[0]$ and to encode the source data in the second sample of each tent map sequence instead of the first. The signal sequences are generated with both backward and forward iteration of the tent map. Specifically, if x_1 and x_2 are the two source letters,

$$\{x_i[n]\}_i = \begin{cases} F^{(-1)}_{s_i[0]}(x_i), & n = 0 \\ F^{(n-1)}(x_i), & n \geq 1. \end{cases} \tag{5.18}$$

One could presumably use a similar strategy for $m > 2$, i.e.,

$$\{x_i[n]\}_i = \begin{cases} F^{(n-l_b)}_{\mathbf{s}_i[0:l_b-1]}(x_i), & n < l_b \\ F^{(n-l_b)}(x_i), & n \geq l_b, \end{cases} \tag{5.19}$$

although the best back propagation signs $\{\mathbf{s}_i[0:l_b-1]\}_i$ and minimum back iteration length l_b are not known at this time. Alternatively, one might avoid these issues by developing estimation algorithms for the case when $\tilde{\mathbf{A}}$ does not have linearly independent columns. In this case there are many $\mathbf{x}[n|n]$ that minimize $\|\mathbf{e}\|^2$ in (5.14), so one would have to develop other criteria for selecting one of the minimizing solutions.

5.3 Error Analysis

One can compute the Cramer-Rao bound for the error variance on the filtered estimates $\{x_i[n|n]\}_i$ by considering the observations to be a random vector $\mathbf{y}$, the sum of a vector function of the unknown parameters $\{x_i[n]\}_i$ and a Gaussian noise vector $\mathbf{w}$,

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) + \mathbf{w}, \tag{5.20}$$

where the ith element of each vector is

$$\begin{aligned} y_i &= y[i], \\ f_i &= \sum_{k=1}^{m} F^{(i-n)}_{\mathbf{s}_k[i:n-1]}(x_k[n]), \\ w_i &= w[i], \qquad \sigma^2_{w[i]} = \sigma_i^2, \end{aligned}$$

for $i = 0, \cdots, n$. The Cramer-Rao bounds for the error variances are given by the diagonal elements of the inverse of the Fisher Information matrix

$$\mathbf{I}_{\mathbf{xy}} = E\left\{\left[\frac{\partial}{\partial \mathbf{x}} \ln p(\mathbf{y}|\mathbf{x})\right]^T \left[\frac{\partial}{\partial \mathbf{x}} \ln p(\mathbf{y}|\mathbf{x})\right]\right\}. \tag{5.21}$$

In this case,

$$\frac{\partial}{\partial x_k} \ln p(\mathbf{y}|\mathbf{x}) = \sum_{i=0}^{n} \frac{(y_i - f_i)}{\sigma_i^2} \frac{\partial f_i}{\partial x_k}, \tag{5.22}$$

where

$$\frac{\partial f_i}{\partial x_k} = (-\beta)^{i-n} \prod_{l=i}^{n-1} s_k[l]. \tag{5.23}$$

Then, the elements the Fisher Information matrix are

$$\begin{aligned}[\mathbf{I_{xy}}]_{ij} &= E\left\{\sum_{k=0}^{n}\frac{(y_k-f_k)}{\sigma_k^2}\frac{\partial f_k}{\partial x_i}\sum_{l=0}^{n}\frac{(y_l-f_l)}{\sigma_l^2}\frac{\partial f_l}{\partial x_j}\right\} \\ &= \sum_k\sum_l E\left\{\frac{w_k w_l}{\sigma_k^2\sigma_l^2}\right\}\frac{\partial f_k}{\partial x_i}\frac{\partial f_l}{\partial x_j} \\ &= \sum_{k=0}^{n}\frac{1}{\sigma_k^2}\frac{\partial f_k}{\partial x_i}\frac{\partial f_k}{\partial x_j} \\ &= \mathbf{a}_i^T\mathbf{S}_n\mathbf{a}_j, \end{aligned} \tag{5.24}$$

where $\mathbf{a}_i$ is the ith column of the matrix $\tilde{\mathbf{A}}$ with the estimates of the signs $\hat{s}_i[l]$ replaced by the actual signs $s_i[l]$. Therefore, if $\mathbf{A}$ denotes the matrix whose ith column is $\mathbf{a}_i$,

$$\mathbf{I_{xy}} = \mathbf{A}^T\mathbf{S}_n\mathbf{A}, \tag{5.25}$$

and the Cramer-Rao bounds on the error variances are

$$\mathbf{K}_{\hat{x}\hat{x}} \geq (\mathbf{A}^T\mathbf{S}_n\mathbf{A})^{-1}. \tag{5.26}$$

One may also arrive at this expression by making the high SNR approximation invoked in the error analysis of Chapter 2, namely that $\{\hat{\mathbf{s}}_i[0:n-1]\}_i \approx \{\mathbf{s}_i[0:n-1]\}_i$. Under this approximation $\tilde{\mathbf{A}} = \mathbf{A}$ and the solution to (5.14) which minimizes $\mathbf{e}^T\mathbf{S}_n\mathbf{e}$ is unbiased and has a covariance matrix

$$E\left\{(\hat{\mathbf{x}}-\mathbf{x})(\hat{\mathbf{x}}-\mathbf{x})^T\right\} = (\mathbf{A}^T\mathbf{S}_n\mathbf{A})^{-1} \tag{5.27}$$

since $\mathbf{K}_{ee} = \mathbf{S}_n^{-1}$.

Example: Two users with stationary AWGN and $\beta > \sqrt{2}$.

This example examines the case where $m = 2$, $\sigma_i^2 = \sigma^2$, and $\beta > \sqrt{2}$. Then, the

Fisher Information matrix (5.25) becomes

$$\mathbf{I_{xy}} = \begin{bmatrix} a & b \\ b & a \end{bmatrix} \frac{1}{\sigma^2}, \tag{5.28}$$

where

$$a = \mathbf{a}_i^T \mathbf{S} \mathbf{a}_i = 1 + \frac{1-\beta^{-2n}}{\beta^2 - 1} \tag{5.29}$$

and

$$\begin{aligned} b &= \mathbf{a}_1^T \mathbf{S} \mathbf{a}_2 \\ &= 1 + \sum_{k=0}^{n-1} \frac{[\mathbf{a}_1]_k [\mathbf{a}_2]_k}{\beta^{2(n-k)}} \\ &\geq 1 - \sum_{k=0}^{n-1} \frac{1}{\beta^{2(n-k)}} \\ &= 1 - \frac{1-\beta^{-2n}}{\beta^2-1} > 0, \qquad \beta > \sqrt{2}. \end{aligned} \tag{5.30}$$

The Cramer-Rao bound matrix is the inverse of the Fisher Information matrix,

$$\mathbf{I_{xy}^{-1}} = \sigma^2 \begin{bmatrix} \frac{1}{a} & -\frac{b}{a^2} \\ -\frac{b}{a^2} & \frac{1}{a} \end{bmatrix} \frac{1}{1 - \left(\frac{b}{a}\right)^2}. \tag{5.31}$$

The diagonal entries of this matrix are lower bounds on the error variances of $\hat{x}_1[n|n]$ and $\hat{x}_2[n|n]$. These are minimized at $n = \infty$ when a is maximum and b is minimum,

$$a \leq \frac{\beta^2}{\beta^2 - 1}, \tag{5.32}$$

$$\frac{b}{a} \geq 1 - 2\beta^{-2}. \tag{5.33}$$

Then,

$$\text{var}\{\hat{x}_i[n|n]\} \geq \sigma^2 \frac{\beta^2}{4}. \tag{5.34}$$

For the case when $\beta = 2$, therefore, the minimum possible error variance after filtering is equal to the noise variance. This example also illustrates that $\mathbf{I_{xy}}$ depends on the data $\mathbf{x}$ since the tightness of the lower bound on b (5.30) depends on how unalike $\mathbf{a}_1$ and $\mathbf{a}_2$ are. Specifically, the bound assumes that each entry of $\mathbf{a}_1$ is the opposite of the corresponding entry of $\mathbf{a}_2$, except for the last entry, which is always equal to 1 for both.

$\square$

5.4 Recursive Estimation

Based on the results of Chapter 2 and their close similarity to the results in this chapter, one might conjecture that a recursive form of the ML estimator exists for the multiple user case, just as it does for the single user case. This conjecture is made even more plausible if one recalls that $x_i[n-1]$ represents the state at time $n-1$ of the dynamic system generating the ith tent map sequence and is all that is needed to predict $x_i[n]$. Furthermore, the optimal estimate $\hat{x}_i[n-1|n-1]$ contains everything in $\{y[0], \cdots, y[n-1]\}$ that is relevant to estimating $x_i[n-1]$. Therefore, since all the noise samples are independent, it appears reasonable to assume that $\hat{x}_i[n-1|n-1]$ contains everything in $\{y[0], \cdots, y[n-1]\}$ that is relevant to estimating $\hat{x}_i[n]$.

Motivated by the above conjecture, this section derives a recursive filter for the multiple access channel in the two user ($m = 2$) case. Specifically, the estimator is given the unbiased estimates $\hat{x}_1[n-1|n-1]$ and $\hat{x}_2[n-1|n-1]$, their covariance matrix

$$\mathbf{K}_{\hat{x}\hat{x}}[n-1] = \begin{bmatrix} \sigma_{\hat{x}}^2[n-1] & \sigma_{12}[n-1] \\ \sigma_{12}[n-1] & \sigma_{\hat{x}}^2[n-1] \end{bmatrix}, \tag{5.35}$$

and an observation

$$y[n] = x_1[n] + x_2[n] + w[n] \tag{5.36}$$

with noise variance σ_w^2. Based on this information, the estimator finds $\hat{x}_1[n|n]$ and $\hat{x}_2[n|n]$. It is unclear whether the recursive filter derived below yields the same estimates as the optimal ML filter (5.17).

5.4.1 Recursive Filter

The strategy of the estimator is the following:

1. Use $\hat{x}_1[n-1|n-1]$ and $\hat{x}_2[n-1|n-1]$ to estimate the sum and difference of $x_1[n]$ and $x_2[n]$ and calculate the uncertainty in these estimates.

2. Combine the estimates from step 1 with the observation $y[n]$ to obtain $\hat{x}_1[n|n]$ and $\hat{x}_2[n|n]$.

In the first step one could also estimate $x_1[n]$ and $x_2[n]$ instead of their sum and difference, but the estimates of the sum and difference have uncorrelated errors, providing a diagonal covariance matrix for step 2. The best estimate of the sum and difference is

$$\hat{x}_1[n|n-1] \pm \hat{x}_2[n|n-1] = F_{\hat{s}_1}(\hat{x}_1[n-1|n-1]) \pm F_{\hat{s}_2}(\hat{x}_2[n-1|n-1]), \tag{5.37}$$

where $\hat{s}_i = \text{sgn}\ \hat{x}_i[n-1|n-1]$. If Δ_i denotes the estimation error in $\hat{x}_i[n-1|n-1]$ and e_1 and e_2 denote the estimation error in the sum and difference, respectively,

$$\begin{aligned} e_1 &= F_{\hat{s}_1}(\hat{x}_1[n-1|n-1]) + F_{\hat{s}_2}(\hat{x}_2[n-1|n-1]) - (x_1[n] + x_2[n]) \\ &\approx \beta - 1 - \beta s_1(x_1[n-1] + \Delta_1) + \beta - 1 - \beta s_2(x_2[n-1] + \Delta_2) - x_1[n] - x_2[n] \\ &= -\beta(s_1\Delta_1 + s_2\Delta_2), \end{aligned} \tag{5.38}$$

where the approximation is the usual high SNR approximation $\hat{s}_i \approx s_i$. Similarly,

$$e_2 \approx -\beta(s_1\Delta_1 - s_2\Delta_2). \tag{5.39}$$

Since the estimates $\hat{x}_i[n-1|n-1]$ are assumed unbiased, e_1 and e_2 have zero mean. The covariance matrix of the Δ_i is $\mathbf{K}_{\tilde{x}\tilde{x}}[n-1]$, so

$$E\{e_1^2\} = 2\beta^2(\sigma_x^2[n-1] + s_1 s_2 \sigma_{12}[n-1]) \equiv K_1[n], \tag{5.40}$$

$$E\{e_2^2\} = 2\beta^2(\sigma_x^2[n-1] - s_1 s_2 \sigma_{12}[n-1]) \equiv K_2[n], \tag{5.41}$$

$$E\{e_1 e_2\} \;=\; 0. \tag{5.42}$$

The uncertainty in the observation $y[n]$ is, of course,

$$E\{w^2[n]\} = \sigma_w^2 \equiv K_3[n]. \tag{5.43}$$

Therefore, step 2 involves solving the matrix equation

$$\underbrace{\begin{bmatrix} \hat{x}_1[n|n-1] + \hat{x}_2[n|n-1] \\ \hat{x}_1[n|n-1] - \hat{x}_2[n|n-1] \\ y[n] \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}}_{\mathbf{A}} \mathbf{x}[n|n] + \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ w[n] \end{bmatrix}}_{\mathbf{e}} \tag{5.44}$$

with

$$\mathbf{K}_{ee} = \begin{bmatrix} K_1[n] & 0 & 0 \\ 0 & K_2[n] & 0 \\ 0 & 0 & K_3[n] \end{bmatrix}. \tag{5.45}$$

A natural quantity to minimize here is $||\mathbf{e}||^2 = \mathbf{e}^T \mathbf{K}_{ee}^{-1} \mathbf{e}$, and the minimizing solution is

$$\mathbf{x}[n|n] = (\mathbf{A}^T \mathbf{K}_{ee}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{K}_{ee}^{-1} \mathbf{y} \tag{5.46}$$

or

$$\begin{aligned} \hat{x}_1[n|n] \;&=\; \frac{1}{2}\left[\frac{\frac{1}{K_1[n]}}{\frac{1}{K_1[n]} + \frac{1}{K_3[n]}} \left(\hat{x}_1[n|n-1] + \hat{x}_2[n|n-1]\right) + \frac{\frac{1}{K_3[n]}}{\frac{1}{K_1[n]} + \frac{1}{K_3[n]}} y[n] \right] \\ &\quad + \frac{1}{2}\left(\hat{x}_1[n|n-1] - \hat{x}_2[n|n-1]\right), &(5.47) \\ \hat{x}_2[n|n] \;&=\; \frac{1}{2}\left[\frac{\frac{1}{K_1[n]}}{\frac{1}{K_1[n]} + \frac{1}{K_3[n]}} \left(\hat{x}_1[n|n-1] + \hat{x}_2[n|n-1]\right) + \frac{\frac{1}{K_3[n]}}{\frac{1}{K_1[n]} + \frac{1}{K_3[n]}} y[n] \right] \\ &\quad - \frac{1}{2}\left(\hat{x}_1[n|n-1] - \hat{x}_2[n|n-1]\right). &(5.48) \end{aligned}$$

5.4.2 Covariance Matrix Updating

To keep the recursion going, one needs to update the covariance matrix $\mathbf{K}_{\hat{x}\hat{x}}[n]$, which will be used to estimate $x_i[n+1|n+1]$. From (5.47) and (5.48), the $\hat{x}_i[n|n]$ are unbiased. Since the elements of $\mathbf{e}$ are all uncorrelated,

$$\sigma_x^2[n] \equiv \operatorname{var}\{\hat{x}_i[n|n]\} = \frac{1}{4}\left[\left(\frac{\frac{1}{K_1[n]}}{\frac{1}{K_1[n]}+\frac{1}{K_3[n]}}\right)^2 K_1 + \left(\frac{\frac{1}{K_3[n]}}{\frac{1}{K_1[n]}+\frac{1}{K_3[n]}}\right)^2 K_3 + K_2\right] \tag{5.49}$$

and

$$\sigma_{12}[n] = \frac{1}{4}\left[\left(\frac{\frac{1}{K_1[n]}}{\frac{1}{K_1[n]}+\frac{1}{K_3[n]}}\right)^2 K_1 + \left(\frac{\frac{1}{K_3[n]}}{\frac{1}{K_1[n]}+\frac{1}{K_3[n]}}\right)^2 K_3 - K_2\right]. \tag{5.50}$$

Finally, for completeness,

$$\mathbf{K}_{\hat{x}\hat{x}}[n] = \begin{bmatrix} \sigma_x^2[n] & \sigma_{12}[n] \\ \sigma_{12}[n] & \sigma_x^2[n] \end{bmatrix}. \tag{5.51}$$

5.5 Simulation Results

To demonstrate the dependence of estimator performance on the actual sequences, the mean square filtering and smoothing error were empirically measured from the ML estimates for two different cases. One of the signals, $x_1[n]$, was the same for both cases. In the first case, $x_2[n]$ was chosen so that its signs alternated between being the same as and different from the signs of $x_1[n]$, i.e., $s_1[n]s_2[n] = (-1)^{n-1}$. In the second case, $x_2[n]$ was chosen so that its signs were as close as possible to being the same as those of $x_1[n]$ while still satisfying the requirement that $\tilde{\mathbf{A}}$ have full column rank, i.e., $s_1[n] = s_2[n]$ for $n \neq 0$, but $s_1[0] \neq s_2[0]$. Figures 5-1 and 5-2 show the averaged results of 100 Monte Carlo simulations for each of the two cases. The noise variance in each case was $\frac{2}{300}$, which corresponds to an SNR of 20 dB, if one defines the signal power to be the sum of the powers of $x_1[n]$ and $x_2[n]$.

As can be seen from the figures, the two cases lead to large differences in per-

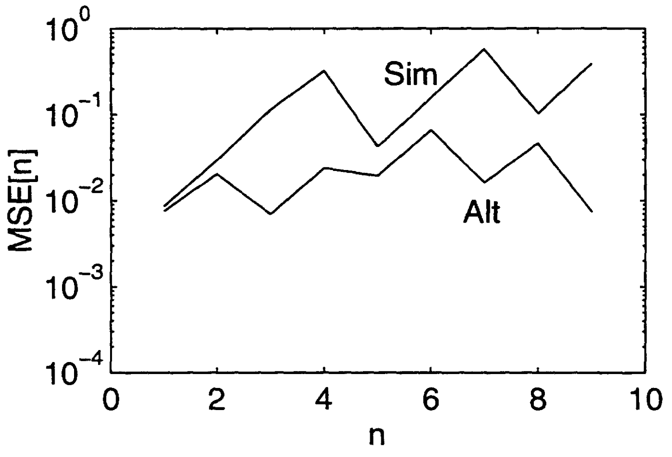

Figure 5-1: Mean Square Error after Filtering for the Two-User Multiple Access Channel. The first user's signal is the same for the two cases shown. The curve marked "Alt" represents the case where the signs of the second user's signal alternates between being the same as and different from the first's. The curve marked "Sim" represents the case where the signs of both users' signals are the same, except for the first sign.

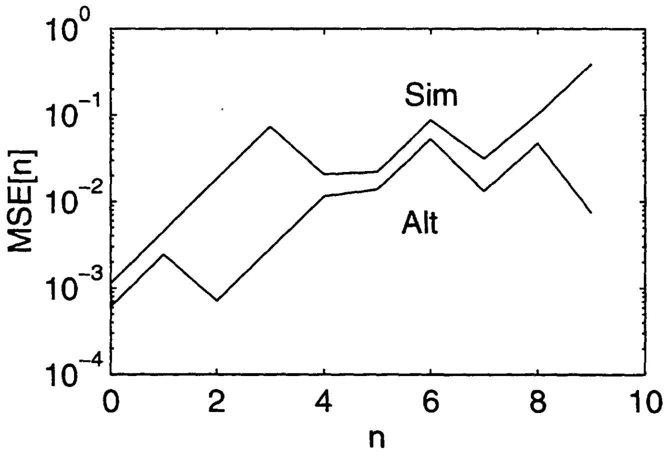

Figure 5-2: Mean Square Error after Smoothing for the Two-User Multiple Access Channel. "Alt" and "Sim" have the same meaning as in Figure 5-1.

formance. The curve marked "Sim" corresponds to a worst case situation, one in which the cross-correlation b (See the example on page 91.) is maximum, given the constraint that $s_1[0] \neq s_2[0]$. Indeed, in the limit of large n, b approaches a in this case and the Fisher Information matrix becomes singular. Thus, the mean square filtering error (Figure 5-1) for this case increases with n. The curve marked "Alt" corresponds to a better case, i.e., lower cross-correlation, but the lower bound on b (5.30) is still not met. It is unclear whether one can construct an example of a case that meets the bound (5.30) for all n. The reader should recall that b represents the cross-correlation of the $\mathbf{a}_i$ and that the entries of the $\mathbf{a}_i$ correspond to *running products* of the $s_i[n]$, not to the $s_i[n]$ themselves. In any event, the results in this chapter are preliminary, and their implications about the potential for chaotic systems in multiple access communications remains an area to be explored in future work.

Chapter 6

Conclusion

Chaotic systems have many potential communication applications. In many of these applications, however, the chaotic signals of interest need to be recovered from corrupted signals which have undergone various forms of distortion. To this end, the work in this thesis involves developing estimation algorithms for chaotic sequences in the presence of additive Gaussian noise, intersymbol interference, and multiple access interference. In addition to developing these algorithms, which may be applicable in many different contexts, this thesis also contains more thorough investigations into the specific application of chaotic systems for modulation and coding of analog source data. Indeed, the applications and algorithms complement each other in the sense that the applications motivate the development of many of the algorithms, while the existence of robust, efficient algorithms suggest which applications may be feasible.

Much of the thesis focuses on the particular chaotic system whose state evolution is governed by the symmetric tent map. As is demonstrated by this thesis, such systems are particularly amenable to analysis since the piecewise linearity of the map can be exploited in state estimation and in performance analysis. Furthermore, in the $\beta = 2$ case tent map sequences have a natural interpretation as a quantization of the initial state $x[0]$.

A major portion of the thesis is devoted to examining tent map sequences and the additive Gaussian noise channel. From the perspective of developing estimation algorithms, Chapter 2 extends the work in [5] to derive a recursive algorithm, which is optimal under the maximum likelihood criterion, for estimating the state variable $x[n]$ in the presence of nonstationary additive white Gaussian noise. The development of this algorithm also suggests a reasonable estimator for the colored Gaussian noise case as well. The performance of this ML estimator is evaluated both analytically and empirically, the main results being that the estimator is asymptotically unbiased at high SNR and has an error variance which decays exponentially with the sequence length before levelling out at a lower threshold.

The existence and performance of the ML state estimator in AWGN suggests a coding or modulation system in which the source data is encoded into the initial state of the chaotic system. This coding system is the subject of Chapter 3, which interprets the system as a twisted modulation system and examines some information theoretic aspects related to its performance. The results in this chapter, although preliminary, demonstrate the potential applicability of chaotic systems, in particular systems whose dynamics correspond to the tent map, to the coding of analog sources. Specifically, the tent map coding system proposed in Chapter 3 outperforms linear modulation coding and optimal M-ary digital coding at high SNR and low bandwidth and is particularly attractive for scenarios in which the SNR is variable or unknown or in which there are multiple SNRs. Hybrid codes combining the tent map and digital codes are also mentioned in this chapter, the exploration of which represents a possible direction of future research.

Motivated by positive results for the AWGN channel, the remainder of the thesis explores the transmission of chaotic signals over the intersymbol interference and multiple access channels. From the perspective of algorithm development, Chapter 4 and Chapter 5 develop state estimation algorithms for tent map sequences which have been corrupted by intersymbol interference and multiple access interference, respectively.

In the ISI case, Chapter 4 presents three algorithms, two of which make explicit use of the deterministic structure of chaotic systems and one of which exploits the sensitivity to initial condition characteristic of chaos. The performance of these algorithms is evaluated empirically, considering the effects of sequence length, noise, and initialization. Although the empirical results provide some general insight into the performance of the algorithms, more work, either empirical or analytical, could provide more thorough characterization of performance and possibly lead to better algorithms. From the applications perspective, Chapter 4 also develops decoders based on the three state estimation algorithms and empirically measures the performance of tent map coding for transmission over the ISI channel.

In the multiple access case, the state estimation algorithm in Chapter 5 is optimal under the maximum likelihood criterion, although the synchronous, uniform amplitude scenario considered is a simplified one. The performance of this algorithm is evaluated both analytically and empirically, including a derivation of the Cramer-Rao bound on error variance. The performance analysis indicates that the estimator is asymptotically unbiased at high SNR and has a variance which depends on the particular tent map sequences. The implication of this dependence in the context of coding applications is not well understood at this point and represents a potentially important direction for future research. Finally, a recursive estimator for the two-user multiple access channel is also developed. It is not known if this estimator is optimal under any meaningful criterion, and this question is also left for future investigations.

Thus, this thesis contains a variety of algorithms for state estimation in the presence of various forms of corruption. These algorithms are potentially useful in many signal modeling and signal synthesis applications, particularly those relating to communication. The application of chaotic systems to coding and modulation is specifically explored, and the results are promising, indicating that although many open questions remain, future work in this area could potentially prove fruitful.

Bibliography

[1] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices, and Groups.* Springer-Verlag, 1988.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., 1991.

[3] Mohammed Dahleh, Munther Dahleh, and George Verghese. Course notes for Dynamic Systems (6.241). This is a set of course notes at the Massachusetts Institute of Technology, 1995.

[4] S. H. Isabelle. *A Signal Processing Framework for the Analysis and Application of Chaotic Systems.* PhD thesis, Massachusetts Institute of Technology, 1995.

[5] H. C. Papadopoulos and G. W. Wornell. Maximum likelihood estimation of a class of chaotic signals. *IEEE Transactions on Information Theory*, 41(1):312–317, January 1995.

[6] O. Shalvi and E. Weinstein. Super exponential methods for blind deconvolution. *IEEE Transactions on Information Theory*, 39(2):504–519, March 1993.

[7] S. H. Strogatz. *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering.* Addison-Wesley Publishing Company, 1994.

[8] J. M. Wozencraft and I. M. Jacobs. *Principles of Communication Engineering.* John Wiley & Sons, Inc., 1965.