GRAPH-BASED DECODERS AND DIVERGENCE-RATE ESTIMATORS FOR DATA-HIDING PROBLEMS

BY

SHANKAR SADASIVAM

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Professor Pierre Moulin, Chair
Professor Richard E. Blahut
Professor Thomas S. Huang
Assistant Professor Todd P. Coleman
Assistant Professor Paris Smaragdis

GRAPH-BASED DECODERS AND DIVERGENCE-RATE ESTIMATORS FOR DATA-HIDING PROBLEMS

BY

SHANKAR SADASIVAM

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Professor Pierre Moulin, Chair
Professor Richard E. Blahut
Professor Thomas S. Huang
Assistant Professor Todd P. Coleman
Assistant Professor Paris Smaragdis

Abstract

In this thesis, we look closely at two fundamental problems that arise within the context of multimedia blind watermark decoding and timing channels steganalysis. The central problem considered, loosely speaking, is that of implementing optimal (or near-optimal) strategies at the receiver, which is typically tasked to perform reliable decoding or detection, depending on the application at hand, in the presence of numerous unavoidable statistical uncertainties that are rather unique to the problem setup. A typical question we will be asking is, "Can we perform reliable decoding of hidden data in spite of the presence of unknown channel parameters?" or "How best can we detect presence of hidden data with unknown, and rather arbitrary, host and observation statistics?" While such questions are naturally relevant from a practical viewpoint, we draw additional inspiration for our study from profound theoretical insights arising from our recent research.

As our solution to the first problem, we propose a new paradigm for blind watermark decoding in the presence of various signal distortion operations. Employing Forney-style factor graphs to model the watermarking system, we cast the blind watermark decoding problem as a probabilistic inference problem on a graph, and solve it via message-passing. We study a wide range of moderate to strong distortions including scaling, amplitude modulation, fractional shift, arbitrary linear and shift invariant (LSI) filtering, and blockwise filtering, and show that the graph-based iterative decoders perform almost as well as if they had exact knowledge of the distortion channel parameters. Other

desirable features of the graph-based decoders include the flexibility to adapt to other types of distortions and the ability to cope with the "curse of dimensionality" problem that seemingly results when the distortion channel parameters' space has high dimensionality. These properties are unlike most blind watermark decoders proposed to date, and close an important computational gap in favor of deploying joint estimation-decoding strategies (shown to be theoretically optimal in our earlier work) to cope with common signal distortions.

For the second problem, we propose new tools for steganalysis of queue-based stegocodes over covert timing channels. We propose a universal estimator for the Kullback-Leibler (KL) divergence-rate between the covertext process and the stegotext process. We empirically illustrate the performance of our estimator on some simple queue-based stegocodes and study its convergence properties.

To Amma and Appa

Acknowledgments

First and foremost, I would like to acknowledge my debt to my adviser, Professor Pierre Moulin, for making graduate school a memorable experience. Through his various incarnations, as a dedicated teacher, articulate orator, optimistic believer, and most importantly an approachable individual willing to listen to pretty much anything under the sun, Pierre seamlessly led me through several years of real world learning and rigor. Thank you, Pierre, for everything!

I would also like to express my sincere gratitude to Professor Todd Coleman and the late Professor Ralf Koetter, without whom the work on graph-based watermark decoding would have been impossible. Discussions with Professors Sean Meyn and Ioannis Kontoyiannis have greatly helped with the development of work on divergence-rate estimators; I look forward to more of their guidance as we will likely continue this research after my graduation.

Many thanks are due to Professors Richard Blahut, Thomas Huang, and Olgica Milenkovic, for serving on my prelim exam committee and providing valuable feedback. I am also deeply indebted to Professor Paris Smaragdis for consenting to serve on my defense committee; I wish he had joined Illinois few years earlier though!

Not to forget – the National Science Foundation (NSF) for funding this research via grants CCR 03-25924, CCF 06-35137, CCF 07-29061 and CCF 08-30776; Professors Bruce Hajek, Venugopal Veeravalli, P. R. Ku-

mar, R. Srikant and Jeff Erickson for the many excellent lectures; the wonderful administrative staff at Beckman and Everitt (Sharon, Paritosh, Laurie, Beth and Sherry in particular) for being extremely helpful and accommodating; Professors David Koilpillai and V. Jayashankar for you were solely behind my humble beginnings at IITM; and the dedicated teaching staff at PS and Satyamurthi for all the invaluable guidance during high school – thank you!

My friends and family, who have been unwavering pillars of support all along, continue to inspire and energize me in ways I cannot imagine. Sincere thanks to the many folks – Dennis, Maha, Tie, Ying, Negar, Amit, Mandar, Shyam to name a few – from whom I frequently sought advice and benefited greatly. I am also indebted to several wonderful people – Jean-Francois, Ibtissam, Yen-Wei, Guillaume, Scott, Rohit, Victor, Nghia, Mert, Yue, Ben, Patrick, Honghai – for making my stay at Beckman over the last five years extremely pleasant. To all my friends from PS, IITM, Illinois, and AFE – this is an extremely long list I dare not to enumerate – a huge thank you for ensuring fun, memorable times spent outside of work! In particular, my housemate, Jayanand, deserves a lot of credit for being extremely patient with me all along! Special thanks are due to Amma and Appa for steadfastly standing behind everything I did, which really meant a lot! Thanks to Sridhar, for being a great friend, and the surprisingly mature younger brother willing to take any number of jabs, frivolous or otherwise, and ensuring my sanity through many difficult times. But for you all, I am sure this journey would have failed long ago.

Finally, I owe it to Chambana (especially the many friendly restaurant owners) for taking real good care of me thousands of miles away from home. So long folks, and thanks for all the fish!

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AM	Amplitude modulation
AWGN	Additive white Gaussian noise
DC-QIM	Distortion compensated quantization index modulation
DFT	Discrete Fourier transform
DWT	Discrete wavelet transform
FFG	Forney-style factor graph
HWR	Host-to-watermark ratio
IDFT	Inverse discrete Fourier transform
i.i.d.	independent and identically distributed
KL	Kullback-Leibler
LSI	Linear shift-invariant
ML	Maximum likelihood
MRF	Markov random field
pmf	probability mass function
pdf	probability density function
QIM	Quantization index modulation
WNR	Watermark-to-noise ratio

Chapter 1

Introduction

The problems that form the subject of this thesis consist of hiding data in cover objects, such as images, videos, audio or possibly even packet transmission instants in a communication network, wherein data-hiding methods offer huge potential to address a variety of modern-day problems. A few applications of data-hiding include copyright protection, fingerprinting, traitor tracing, content authentication, signature verification, media forensics and steganography. Data-hiding research is relatively mature today, and plenty of resources, including survey papers [1–3], textbooks [4–8], and special issues of journals, such as *IEEE Transactions on Signal Processing* supplements on secure media (October 2004 and February 2005), *Proceedings of the IEEE* special issue on Digital Rights Management (June 2004), *IEEE Signal Processing Magazine* (September and November 2003), etc., offer a detailed overview of topics in this area.

1.1 Theory and Practice

For almost all of its existence, data-hiding research has proceeded along two distinct, nonintersecting trails of thought. Many specialized, practical algorithms that have been developed for various data-hiding setups lack sound theoretical backing and possess critical weaknesses. On the other hand, fundamental theoretical analyses of data-hiding systems, for cases where they exist, often overlook important practical issues and therefore have little commercial relevance. This disturbing lag between theory and practice can in fact be thought of as a primary

reason for the lack of a "foolproof" watermarking (or fingerprint) encoding/decoding, and steganography/steganalysis algorithm till date.

Much of the theoretical advances in data-hiding, including computation of channel capacity for various problems of interest, have relied on a crucial observation connecting data hiding to communication theory [9]. Most data-hiding problems can be seen as an instance of communication over a noisy channel, with a side-informed transmitter (and possibly receiver). All the same, practical data-hiding channels differ markedly from the simplistic noise models typically assumed to derive information-theoretic insights in the standard communication setup. In fact, it is frequently possible for channels in the data-hiding case to possess one or more of memory, nonlinearity, time-variability, nonstationary and nonergodic properties, thereby rendering many theoretically well-motivated data-hiding algorithms (typically developed only for simplified relaxations of the actual problem of interest) blatantly unsuitable for direct use in real world applications.

The above discussion forms the basis for our choice of problems studied in this thesis. Both problems chosen admit elegant and analytically tractable solutions, if only we had the luxury to ignore (or had exact knowledge of) the bad world of incredibly rich noisy channel models and/or complicated signal statistics intrinsic to our system setup. This thesis forgoes that luxury, and attempts to make the most of theory and practice to offer solutions that will eventually matter to commercial applications. We accomplish this with help from novel, fundamental techniques from probabilistic inference and statistical estimation theory.

Let us now look at some problem specifics.

1.2 Two Problems

1.2.1 Robustness of Good Blind Watermarking Codes

The advent of the internet and other public information sharing networks has given rise to a multitude of applications where multimedia watermarking plays (or has the potential to play) an important role. Some of these applications involve the presence of an adversary attempting to disrupt reliable communication of the information of interest to the receiver. In particular, it has been observed that simple signal distortion operations (e.g., scaling, amplitude modulation, global or locally varying shifts (warping), filtering, gamma correction, geometric (spatial) transformations such as rotation, zooming, etc.) can have a catastrophic impact on the performance of the decoder [10–13], usually measured via the probability of correctly decoding hidden data (i.e., the watermark). Also, the decoder need not, in general, have access to the original unmarked data, a scenario commonly referred to as *blind data-hiding* [3]. Good performance can theoretically be obtained using *binning schemes* [3, 14] such as quantization index modulation (QIM) and spread transform dither modulation (STDM), but those schemes *appear* to be brittle against common distortion operations, as evidenced by a review of the literature. This challenge is, to date, one of the most difficult and, hence, least resolved problems in the field.

Broadly, three types of solutions have been proposed in the literature to address this problem: embedding watermarks in an appropriate distortion invariant domain [13, 15–19], embedding pilots (synchronization sequences) to help with inverting the distortion(s) [20–26] and employing a joint estimator-decoder that decodes the watermark and estimates the distortion channel parameters simultaneously [10–12, 27–29]. With the exception of [17] in a noise-free scenario, most

invariant based approaches are tailored to handling simple (e.g., pure scaling or pure shift) distortions, and so they offer little or no insight into handling other complex scenarios. Pilots are not information bearing; while convenient, they reduce the embedding efficiency and are theoretically suboptimal [30]. A theoretically superior approach is to design a code that lends itself to resynchronization, without wasting resources in communicating training sequences that are not information-bearing [24,25,31]; the joint estimator-decoders mentioned above can be classified under this category.

As an example of this approach, some of the best results to date for the blind embedding problem have been obtained by Balado *et al.* [10]. They explore the use of the expectation-maximization (EM) algorithm for simultaneously decoding messages and estimating scale and delay parameters. In more recent work [29], they explored the use of phase locked loops as an alternative to the EM algorithm. Unfortunately, poor performance is obtained when the channel distortion is moderate or large.

The lack of any formidable breakthrough in combating simple distortion operations on blind watermarking systems, in spite of several years of research, inevitably leads one to wonder whether the poor performance is due to some fundamental decoding limitation introduced by the distortion channel, or is it merely suboptimal design? Our recent work relating to this question uncovered a series of interesting revelations, which we briefly summarize below.

1. The joint estimator-decoder is *asymptotically optimal* in the sense that it has the same decoding error exponent as that of a *coherent decoder* which knows the distortion channel exactly [32, 33]. In other words, there exists a sequence of decoding rules g_n so that,

under certain regularity and smoothness conditions,

$$\limsup_{n\to\infty} \max_{\theta'\in\Theta} \frac{1}{n} \log \frac{P_e(\theta', g_n)}{P_e(\theta)} = 0, \tag{1.1}$$

where n is the dimensionality of the real-valued host sequence, Θ is the parameter space and $P_e(\theta)$ is the decoding error probability of the coherent decoder (that knows θ).

2. It is possible to obtain tight confidence bounds on the estimation accuracy of distortion channel parameters at the receiver [34]. Specifically, the estimation error covariance matrix of the distortion parameters, $\mathrm{cov}_\theta[\hat{\theta}(\mathbf{y})]$, of any unbiased estimator can be lower-bounded as

$$\mathrm{cov}_\theta[\hat{\theta}(\mathbf{y})] \succeq \left(\sum_{i=1}^{\rho} \mathbb{E}_\theta \Big[-\nabla_\theta^2 \log p^{(i)}_{\theta,m=0}(\mathbf{u}^{(i)}) \Big] \right)^{-1}, \tag{1.2}$$

where ρ is the number of independent components of the received signal $\mathbf{y}$, and $\{\mathbf{u}^{(i)}\}_i$ denote a 'smart' partition of $\mathbf{y}$, with the dimensionality of each element (i.e., $\mathbf{u}^{(i)}$) restricted to two. More details can be found in [34].

3. Further, it is possible to explicitly identify receivers that get close to the confidence bound in (1.2). Specifically, a joint estimator-decoder that has access to huge computational power, and can exhaustively search over the entire distortion parameter space, almost achieves the bound in (1.2) for a wide variety of distortions (including scaling, shift, and any arbitrary linear shift-invariant transformation) [34].

These are important results for an obvious reason: together, within statistical reason, they establish the joint feasibility of accurate distortion channel parameters' estimation and reliable blind decoding of the hidden message, thus answering a longstanding puzzle in the watermarking community. However, favorable optimality properties notwithstand-

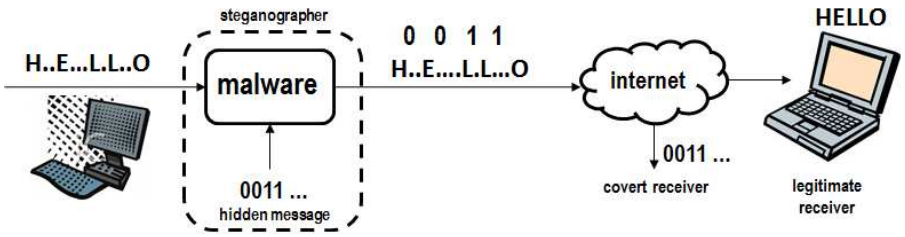


Figure 1.1: A simple trojan horse (malware) + timing channel setup. Here, an even (resp. odd) number of time slots between two consecutive letters encodes bit 0 (resp. 1).

ing, the practicality of estimator-decoders remains a serious concern due to the need for searches over a possibly large parameter space [34]. It is not clear how one could come up with efficient computational methods, that possibly trade off speed against accuracy, for handling various types of signal distortions. We explore this topic further in Chapter 2.

1.2.2 Universal Steganalyzers for Timing Channels

Timing channels are covert channels where information is encoded into the timings of packets sent by a transmitter [35, 36]. They can generally coexist with asynchronous communication networks, or when data sources transmit packets at irregular time instants. Timing channels can be used (e.g., by military and intelligence agencies) to discreetly communicate over public networks. On the other hand, timing channels can also be designed by an adversary for the same purpose. The adversary might be a byzantine user in a computer network, or a hacker who has gained unauthorized access to a computer and leaks out information residing in the system (e.g., passwords, encryption keys, or other sensitive data). The latter problem is also referred to as *data exfiltration*, and a simplified version of the same is shown in Figure 1.1. One can think of the transmitter as being involved in an interactive communication session (e.g., instant messaging). The steganographer has access to these packets (possibly via malware in-

stalled on the transmitter's computer) and modulates the time between keystrokes to covertly communicate a message of importance to the corresponding covert receiver. The modulation is done in such a manner that an even (resp. odd) number of time slots between two consecutive packets represent bit 0 (resp. bit 1). If an interarrival time matches the desired bit (w.r.t. parity), the steganographer does not modify it; otherwise he delays the incoming packet by one time slot. He does not modify or delete packets, nor does he need to know the packet contents.

As a mandatory requirement, regardless of whether intent is malicious or otherwise, users of covert timing channels desire that the very presence of this transmission remains hidden from the network administrator (and anyone else). Steganography in timing channels deals with the problem of designing communication schemes that are statistically undetectable, relative to the default communication pattern in the network (one in which the covert channel is not exploited).

The dual problem of steganography is steganalysis, that is, detection of hidden information within a dataset. In applications of covert communication, steganalysis assumes a great deal of significance, and its study is inseparable from that of steganography itself. It would be unsafe for the transmitter to be unaware that its communication is detectable using advanced statistical methods; dually, the steganalyzer could be lulled into a false sense of security stemming from his ability to detect a few rudimentary steganographic operations. Steganalysis presents significant advantages over alternative adversary disruption strategies such as jamming that disrupt normal packet traffic and cause latency and packet transmission errors. Hence, detecting covert communication via steganalysis is a less disruptive and more desirable approach to dealing with network intrusions. Detection has been considered a difficult problem in the timing channel literature, and in fact only a

few papers have broached the subject [37, 38].

If one has access to (or is able to accurately estimate) the joint statistics of interpacket times, both with and without hidden data, the task of testing an observed dataset (i.e., interpacket times) for presence of hidden information is straightforward. The optimum thing to do (w.r.t. minimizing Bayesian risk) would be to evaluate the standard log-likelihood ratio corresponding to the given dataset, and declare presence or absence of hidden data as appropriate. Unfortunately, for most interesting situations (that typically arise when the encoder is reasonably smart), this is a tall order, even if we have access to large amounts of labeled covertext and stegotext[1] data. This is because smart encoders (e.g., see queue-based encoders of Section 3.3) try to maximize their undetectability by intelligently introducing (possibly long range) correlations in the stegotext, thereby inducing a "curse of dimensionality" problem for the estimation of stegotext joint statistics from finite data. Although typically less pronounced than with stegotext statistics, correlations can also exist in covertext data, leading to similar estimation difficulties.

Given these practical issues, we ask the following questions: In the absence of likelihood-ratio tests, what is the next best option? Is it possible to devise universal detection rules for the above problem setup? Can we guarantee anything (e.g., consistency, computational complexity) for universal detection rules, if they exist?

[1] modified covertext, containing hidden information

1.3 About This Thesis

1.3.1 Main Contributions

The contributions of this thesis are two-fold.

First, to address questions raised in Section 1.2.1, Chapter 2 introduces a *practical* computational framework for decoding in the presence of several signal distortions using *graphical models* for the host signal, the watermarking code, and the distortion channel.[2] These models appear to be particularly appropriate for watermarking of media signals because the underlying probabilistic models are local, and inference problems such as watermark decoding can be solved using iterative belief propagation (a.k.a. message-passing) algorithms. We will illustrate our approach with numerical results for various moderate to strong intensity scaling, amplitude modulation, fractional shift, and other (global and blockwise) LSI filtering operations.

Second, to address the questions of Section 1.2.2, Chapter 3 presents a goodness-of-fit test for performing steganalysis in timing channels. The test operates via an estimate of the Kullback-Leibler (KL) divergence-rate between covertext and stegotext processes, a quantity fundamental to both the design and analysis aspects of stegocodes for timing channels. As a means to the end, we propose a novel match-length based estimator for the KL divergence-rate between covertext and stegotext processes, and study its convergence properties using various queue-based stegocodes for timing channels.

[2]The reader is referred to the books by Lauritzen [39], Pearl [40] and Frey [41] as well as the articles [42–44] for a general introduction to graphical models.

1.3.2 Organization

The rest of the thesis is organized as follows. Chapters 2 and 3 present a detailed study of the problems introduced in Sections 1.2.1 and 1.2.2 respectively. Chapter 4 discusses some promising future directions of our work. Finally, Appendix A presents details of a parameter estimation algorithm used in Chapter 2.

Chapter 2

Distortion-Resilient Blind Watermark Decoding

This chapter is organized as follows. Following some notation and definitions in Section 2.1, we provide our system model and motivate the distortion-resilience problem further in Section 2.2. Graphical models are introduced in Section 2.3 and the proposed decoding algorithm is presented in detail in Section 2.4. Experimental results corresponding to various distortion models are reported in Section 2.5. We briefly summarize the chapter in Section 2.6, following a discussion highlighting some caveats associated with our algorithm.

2.1 Notation

The following will remain in effect throughout this chapter.

- Both random variables and their realizations are denoted by lowercase letters; the context will make the notation clear on all occasions.
- Boldface letters denote sequences of real numbers, e.g., $\mathbf{x} = (x_i)_{i \in \mathscr{I}}$, where $\mathscr{I}$ is a finite set.
- $\|\mathbf{x}\| \triangleq \sqrt{\sum_{i \in \mathscr{I}} x_i^2}$, the Euclidean norm of $\mathbf{x}$.
- The probability density function (pdf) of $\mathbf{x}$ is $p_{\mathbf{X}}$; $p_{\mathbf{X}}(\mathbf{x})$ (or in short $p(\mathbf{x})$) denotes the pdf evaluated at a point $\mathbf{x}$.
- The probability density function (pdf) of $\mathbf{x}$ conditioned on $\mathbf{y}$ is $p_{\mathbf{x}|\mathbf{y}}$; $p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})$ (or in short $p(\mathbf{x}|\mathbf{y})$) denotes the conditional pdf evaluated at a point $\mathbf{x}$.

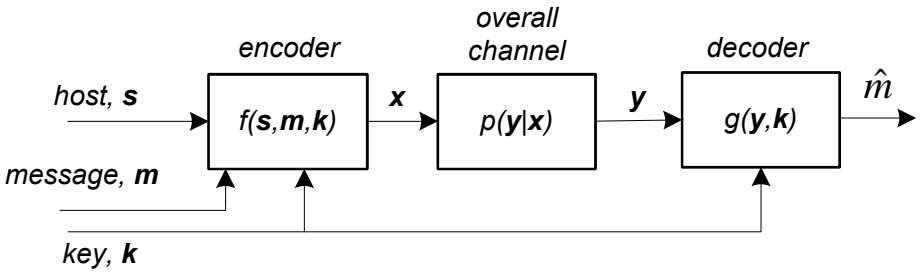


Figure 2.1: Communication model for watermarking.

- $\mathcal{N}(\mu, \sigma^2)$: Gaussian distribution with mean μ and variance σ^2.
- $\delta(\cdot)$: Dirac impulse.

2.2 System Model

A fairly general communication model for watermark decoding is depicted in Figure 2.1. A length-n_b message, such as a digital signature, $\mathbf{m} = \{m_i, 1 \leq i \leq n_b\} \in \{0,1\}^{n_b}$ is embedded in a multidimensional real valued host sequence $\mathbf{s} = \{s_i, i \in \mathscr{I}\}$, aided by side information $\mathbf{k} \in \mathbb{R}^{|\mathscr{I}|}$ shared with the receiver. The signal after embedding is denoted by $\mathbf{x} = f(\mathbf{s}, \mathbf{m}, \mathbf{k})$ and referred to as the watermarked (or simply the marked) signal. In some cases, no embedding takes place (say when $\mathbf{m} = \emptyset$), and the encoding function f simply reproduces $\mathbf{s}$. The receiver does not observe $\mathbf{x}$ directly. It only observes the output of an *insecure channel* modeled by a conditional distribution $p(\mathbf{y}|\mathbf{x})$. For instance $p(\mathbf{y}|\mathbf{x})$ could be a simple memoryless channel, such as an additive white Gaussian noise (AWGN) channel. But the insecure channel need not be memoryless or even causal (see the examples of Section 2.5).

The encoder is assumed to act blockwise on the host: the embedding function $f(\cdot)$ factors into n_b embedding functions f_b that act independently on nonoverlapping b-dimensional host subblocks, where $b = |\mathscr{I}|/n_b$. More precisely, the host, the marked sequence and the key

may be partitioned as

$$\mathbf{s} = \left(\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(n_b)}\right), \tag{2.1}$$

$$\mathbf{x} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n_b)}\right), \tag{2.2}$$

$$\mathbf{k} = \left(\mathbf{k}^{(1)}, \mathbf{k}^{(2)}, \ldots, \mathbf{k}^{(n_b)}\right), \tag{2.3}$$

where

$$\mathbf{x}^{(i)} = f_b(\mathbf{s}^{(i)}, m_i, \mathbf{k}^{(i)}) \in \mathbb{R}^b, \quad 1 \le i \le n_b. \tag{2.4}$$

The function $f_b(\cdot)$ in (2.4) is the scalar DC-QIM embedding function acting on each component of $\mathbf{s}^{(i)}$, i.e.,

$$x_j^{(i)} = (1-\alpha)s_j^{(i)} + Q_{m_i}(\alpha s_j^{(i)} - k_j^{(i)}) + k_j^{(i)}, \tag{2.5}$$

where $i = 1, 2, \ldots n_b$; $j = 1, 2, \ldots b$; $x_j^{(i)}$ is the j^{th} component of $\mathbf{x}^{(i)}$; $\alpha \in (0, 1]$ is the distortion-compensation (Costa) parameter; $Q_m(s) = Q\left(s - (-1)^m \frac{\Delta}{4}\right) + (-1)^m \frac{\Delta}{4}$ is the shifted quantizer associated with bit $m \in \{0, 1\}$; and $Q(\cdot)$ is the prototype scalar uniform quantizer with step size Δ. As implied by (2.5), the same bit m_i is embedded in each component of $\mathbf{s}^{(i)}$, thereby inducing a rate $R = 1/b$ repetition code within each subblock $\mathbf{x}^{(i)}$. The host-to-watermark ratio (HWR) of the embedding process is given by

$$HWR \triangleq \frac{1}{|\mathscr{I}|} \frac{\|\mathbf{s}\|^2}{D_1}, \tag{2.6}$$

where $D_1 = \frac{\Delta^2}{12}$ is the watermark power[1]. The setup of Figure 2.1 is applicable both to authentication problems (in which case the embedding rate R is typically small) and to data hiding (where R is large).

The receiver has access to the received signal $\mathbf{y}$ and side information

[1]For a wide range of host signals, the dynamic range of the host is much larger than the quantization step size Δ, enabling us to approximate the watermark embedding distortion per sample to be uniformly distributed in the interval $[-\frac{\Delta}{2}, \frac{\Delta}{2})$.

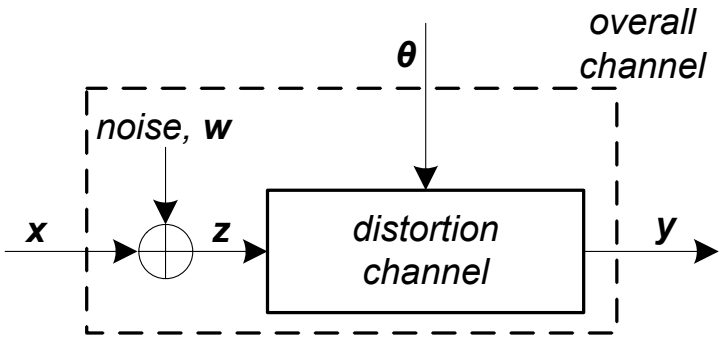


Figure 2.2: Model for overall channel: noise + distortion.

k and produces an estimate of **m**. We assume that the receiver knows the *type* of signal distortion, and the statistical models of the host and the distortion channel, but not necessarily the exactchannel parameters. The side information **k** may be a cryptographic key, but can also be used to convey information about **s** to the receiver. A *blind receiver* is not given access to the host **s**. For our purposes, we assume **k** is independent of **s** and, as diagrammed in Figure 2.2, model the insecure channel as the cascade of an additive Gaussian noise channel followed by a distortion transformation[2] parameterized by θ. In the simplest setting, the dimensionality of θ does not depend on the signal size $|\mathscr{I}|$; more generally, θ may be a sequence $\theta(i), 1 \leq i \leq |\mathscr{I}|$, that exhibits temporal coherence properties, i.e., it is slowly varying, with occasional jumps. A hypothetical decoder that is informed of the values of these parameters is a *coherent decoder*; a decoder that does not is a *noncoherent decoder*. We shall be interested in constructing good noncoherent decoders, and in estimating the noncoherent decoding penalty.

[2]Although the separation is blurry, throughout this thesis, we will use the terms 'noise' and 'distortion' to distinctly refer to, respectively, the additive noise block at the front end of our overall channel, and the parameteric signal transformation that follows it.

2.3 Graphical Models

A particularly exciting opportunity in the watermark decoding problem is the possibility to combine the Bayesian paradigm for optimal decision making with inference techniques on graphical models [40–42]. For instance, classical Kalman filtering (or Kalman smoothing) may be interpreted as an instance of probabilistic inference in a special Gaussian graphical model. The use of Bayesian recursive filters in lieu of Kalman filters is a natural extension to this technique to nonlinear state-space models.

The opportunity of graphical models in the context of watermark decoding consists of a way to embed a message with some redundancy in a host which can exhibit long range dependencies among the signal components. The final estimation can then be organized in such a way that the Bayesian estimator and an estimator for the data redundancy iteratively solve the probabilistic inference problem. This iterative approach to estimating data in noisy environments has been successful in data transmission and is dubbed the "turbo" principle. In our context we want to fully exploit the power of this approach even in hostile and difficult environments as would be constituted by an active distortion transformation on the decoding scheme. As such, our approach is motivated by work on the probability propagation (sum-product) algorithm for iteratively decoding error-correcting codes such as turbo codes [41]. Until recently, optimal decoding even on Gaussian channels was thought to be intractable. However, it turns out that probability propagation in a graphical model describing the code solves the problem for practical purposes.

The power of this approach is even more apparent in higher (e.g., two) dimensional data sets as they naturally appear in watermarking of images or video sequences. In other words, the "curse of dimensionality"

is a problem that is efficiently addressed in graphical models. In fact, one can argue that graphical models were specifically invented to cope with inference problems in high-dimensional setups. In this case, a graphical model may be used to estimate a distortion or alteration in the properties of the host data. The essential trick is to find a decomposition of the posterior probability density function such that estimation and hypothesis testing has a tractable structure. The generic problem in our problem setup would be one where the adversary has K possible transformations (the first one being time warping, possibly using a multiscale representation for the warping process; the second one might be an amplitude modulation, again using a multiscale representation for the envelope; etc.). The key to coping with the dimensionality of such a model is to find (or model) a factorization of the probability density as is done, for example, in factor graphs [41, 42]. Once this is done, powerful inference algorithms such a the sum-product algorithm can effectively construct excellent approximations to the global objective function [41, 42], and together with a powerful, interleaved code that protects the embedded data, we can obtain an efficient scheme for data embedding. Moreover such a scheme is computationally feasible due to its inherent divide-and-conquer philosophy, thereby making parallelized implementation approaches feasible. This approach has revolutionized much of communications in the last few years and we believe that it holds the potential to give similarly significant and practical improvements for the watermark decoding problem; experimental evidence presented later on in this chapter will add further credence to this belief. We will study the simple (and yet powerful) message-passing algorithm in the next section.

2.4 Message-Passing Decoding Algorithm

The application of the message-passing algorithm to find an approximate maximizer (or even better, an exact maximizer, e.g., if the graph

is a tree) to the desired objective function (here $p(\mathbf{m}|\mathbf{y})$) is a three step procedure:

1. **Graph**: Write down the Forney-style factor graph (FFG) corresponding to the system at hand [42, 45]. To do this, we need to first factorize the joint probability distribution of all variables involved in the system using the appropriate conditional probabilities, thus resulting in several factors that typically depend on a rather small subset (e.g., cardinality two or three) of all variables in the system. The construction of the corresponding FFG is then straightforward, and follows the following rules:

   (a) A unique node for every factor.

   (b) A unique edge (connecting two nodes) or half edge (connected to only one node) for every variable.

   (c) The node representing some factor g is connected with the edge (or half edge) representing some variable x if and only if g is a function of x.

   This is best explained through an example. For ease of exposition, let us momentarily assume that the distortion transformation in Figure 2.2 does nothing (i.e., $\mathbf{y} = \mathbf{z}$). Further, let us assume that all signals are one dimensional, the embedding rate is $R = 1/|\mathscr{I}|$, and we use the following Markov random field based pdf to model the host $\mathbf{s}$:

$$p(\mathbf{s}) = \frac{1}{Z}\prod_{i\in\mathscr{I}} \psi(s_i, \sigma_{sa}^2)\,\psi(s_i - s_{iE}, \sigma_{sb}^2), \tag{2.7}$$

   where $\psi(s, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{\frac{-s^2}{2\sigma^2}\}$ is the Gaussian pdf, $Z, \sigma_{sa}^2, \sigma_{sb}^2$ are constants and the index iE refers to the host sample located to the east (right) of i (cycle around if necessary since we assume periodic extensions of the images). Here, the joint pdf of

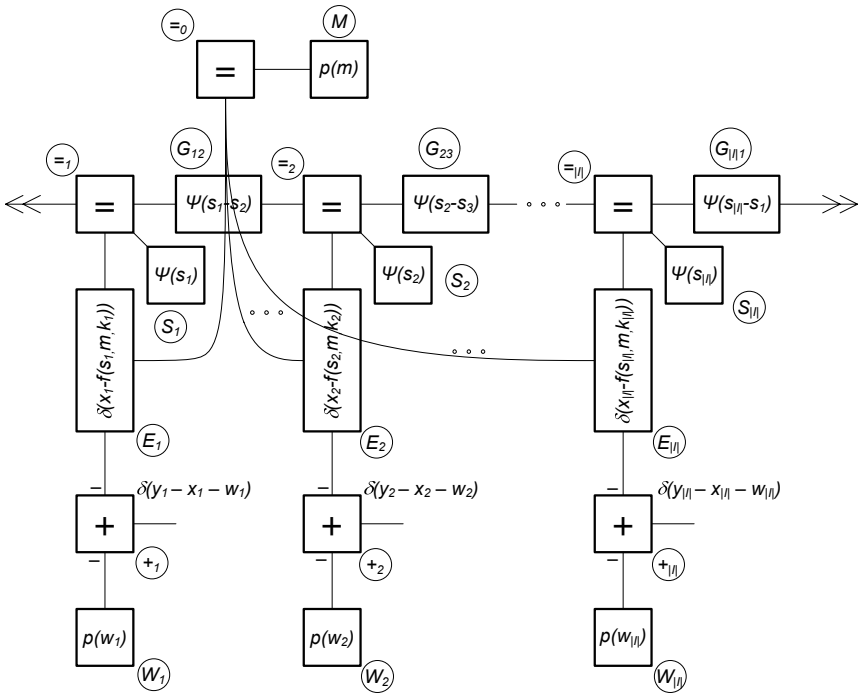


Figure 2.3: Factor graph toy example for Section 2.4. Note that edges marked with double arrows are connected to each other, thus creating a loopy graph modeling host statistics. Text within circles denotes node labels, used for convenience.

all system variables factors as follows:

$$p(m,\mathbf{s},\mathbf{x},\mathbf{w},\mathbf{y}) = p(m)\underbrace{\frac{1}{Z}\prod_{i\in\mathscr{I}}\psi(s_i,\sigma_{sa}^2)\,\psi(s_i-s_{iE},\sigma_{sb}^2)}_{p(\mathbf{s})} \cdot \underbrace{\prod_{i\in\mathscr{I}}\big(p(x_i|s_i,m)p(w_i)p(y_i|x_i,w_i)\big)}_{p(\mathbf{x}|\mathbf{s},m)p(\mathbf{w})p(\mathbf{y}|\mathbf{x},\mathbf{w})}, \tag{2.8}$$

and the factor graph corresponding to the joint pdf of $(m,\mathbf{s},\mathbf{x},\mathbf{w},\mathbf{y})$ is given in Figure 2.3. The "+" and "=" blocks are special factor nodes representing the zero-sum and equality constraint nodes respectively; the reader is referred to the tutorial paper by Loeliger [45, Section 2] for a lenghtier introduction to the zero-sum and equality constraint nodes. For the sake of brevity, we omit

the second argument to $\psi(\cdot)$ in the graph.

2. **Schedule:** Pick a schedule according to which the messages corresponding to each edge on the graph will be updated. There are two messages corresponding to each edge – we shall refer to these as the forward and backward messages. The forward messages are updated in the 'forward pass' of the message updating step, and similarly for the backward. For all of our experiments, we stick to the 'logical' forward schedule as suggested by the block diagram in Figure 2.1, left to right; i.e., in Figure 2.3, the forward messages are propagated from the host level to the marked signal level, and then further downwards. The whole forward sequence of propagation is then retraced to complete the 'backward pass,' thereby completing one iteration of a full message update. A valid schedule for the factor graph of Figure 2.3 is given in Table 2.1.

3. **Update and Iterate:** This one is the message updating step, and is rather straightforward. All messages, forward and backward, corresponding to every edge in the graph are initialized to 1 in the beginning. Thereafter, in accordance with the chosen schedule, update of an outgoing message $\mu_n(t_n)$ at an arbitrary node with factor $\lambda(\cdot)$ (i.e., λ could be any of the individual factors in (2.8)) and incoming messages $\mu_1(t_1), \mu_2(t_2), \ldots \mu_{n-1}(t_{n-1})$ (see Figure 2.4) takes the following general form:[3]

$$\mu_n(t_n) = \int_{\mathbb{R}^{n-1}} \left(\prod_{i=1}^{n-1} \mu_i(t_i) \right) \lambda(t_1, t_2, \ldots, t_n)\, dt_1 \ldots dt_{n-1}. \quad (2.9)$$

[3]Note that the factorization process, as done in (2.8), typically results in factors containing only a small number of variables, thereby making the computation in (2.9) tractable.

Table 2.1: A message update schedule (one forward pass and one backward pass) corresponding to the factor graph of Figure 2.3. Here, for arbitrary U and V, $\mu_{U\to V}$ denotes the message directed from node labeled U to node labeled V.

Forward Pass	**Backward Pass**
Update $\mu_{M\to=_0}$	**for** $i = |I|$ to 1
	Update $\mu_{+_i\to W_i}$
for $i = 1$ to $|I|$	**end**
Update $\mu_{=_0\to E_i}$	
end	**for** $i = |I|$ to 1
	Update $\mu_{+_i\to E_i}$
for $i = 1$ to $|I|$	**end**
Update $\mu_{S_i\to=_i}$	
end	**for** $i = |I|$ to 1
	Update $\mu_{E_i\to=_i}$
Update $\mu_{=_1\to G_{|I|1}}$	**end**
Update $\mu_{G_{|I|1}\to=_{|I|}}$	
for $i = 1$ to $|I|-1$	**for** $i = |I|-1$ to 1
Update $\mu_{=_{i+1}\to G_{i,i+1}}$	Update $\mu_{=_i\to G_{i,i+1}}$
Update $\mu_{G_{i,i+1}\to=_i}$	Update $\mu_{G_{i,i+1}\to=_{i+1}}$
end	**end**
	Update $\mu_{=_{|I|}\to G_{|I|1}}$
for $i = 1$ to $|I|$	Update $\mu_{G_{|I|1}\to=_1}$
Update $\mu_{=_i\to E_i}$	
end	**for** $i = |I|$ to 1
	Update $\mu_{=_i\to S_i}$
for $i = 1$ to $|I|$	**end**
Update $\mu_{E_i\to+_i}$	
end	**for** $i = |I|$ to 1
	Update $\mu_{E_i\to=_0}$
for $i = 1$ to $|I|$	**end**
Update $\mu_{W_i\to+_i}$	
end	Update $\mu_{=_0\to M}$

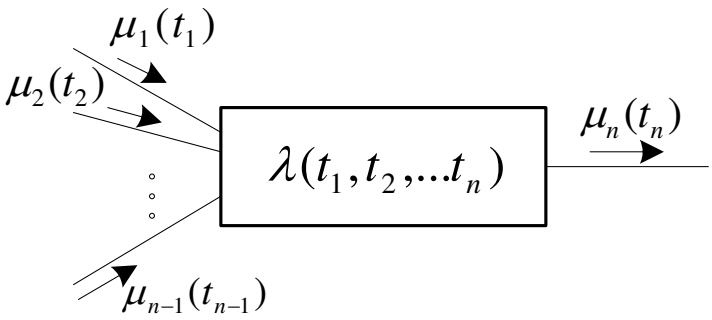


Figure 2.4: Message update at factor node.

This step is therefore purely about updating messages at each node according to the schedule chosen above, until some convergence criterion is met.[4] In this work, since we are interested in decoding the watermark, we iterate until the decoder's output stabilizes for τ iterations, where τ is a small number (< 10) chosen apriori. Choosing $\tau = 1$ will suffice if the underlying FFG is nonloopy, but this is typically not the case in our problems of interest.

Note that in the special case when λ corresponds to an equality constraint node, (2.9) simplifies to:

$$\mu_n(t_n) = \prod_{i=1}^{n-1} \mu_i(t_n). \tag{2.10}$$

Another frequently encountered scenario is when λ is the zero-sum constraint node and $n = 3$. Here, (2.9) simplifies to a simple convolution:

$$\mu_3(t_3) = \int_{\mathbb{R}} \mu_1(t_1)\mu_2(t_3 - t_1)dt_1. \tag{2.11}$$

[4]These algorithms are generally known to converge to local extrema, and convergence to globally optimum solutions is difficult to guarantee, especially when the underlying graph is heavily loopy (as is the case in our problems of interest), and/or when the messages are rather irregular, e.g., multimodal (again, as is the case in our problems of interest).

A remark on the actual computation of the integral involving continuous valued variables in (2.9): Several ways to do this include simple discretization of continuous valued messages, parametric updates, and particle methods wherein messages are represented as lists of samples. A detailed summary of the various methods can be found in Dauwels' doctoral thesis [46]. In this work, all messages are computed in a discretized fashion, excepting when explicit parametric updates are possible (more details regarding this in Section 2.5).

2.5 Experimental Results

All our experiments have been done on two dimensional (256×256) signals, including synthetic and photographic images. We model the host $\mathbf{s} = \{s_i, i \in \mathscr{I}\}$ as a Gaussian Markov random field (MRF) with first-order neighborhoods as follows:

$$p(\mathbf{s}) = \frac{1}{Z_1} \prod_{i \in \mathscr{I}} \psi(s_i - \mu_s, \sigma_{sa}^2)\, \psi(s_i - s_{iE}, 4\sigma_{sb}^2)\, \psi(s_i - s_{iN}, 4\sigma_{sb}^2), \quad (2.12)$$

where $\mathscr{I}$ again denotes the pixel index set; $\psi(s, \sigma^2)$, as defined earlier, is the Gaussian pdf; and s_i, s_{iE} and s_{iN} denote the pixel intensities at location i, one pixel to the east, and one north of (or directly above) i, respectively (cycle around if necessary).[5] Z_1 is the normalization factor for the pdf. We will also stick to an information embedding rate of $R = 1/8$ (one bit is embedded into each 4×2 subblock of the host) throughout as it helps with making comparisons from various experiments. We will focus on five types of distortion models described below:

(M1) **Amplitude scaling:** $\mathbf{y} = \theta\mathbf{z}$, where in our experiments, θ is a Gaussian random variable with mean 1 and standard deviation 0.2. Alternatively, we may have $\theta_{\min} \leq \theta \leq \theta_{\max}$ in which case the

[5] If the model parameters are not known a priori at the receiver, they should be estimated. We discuss this in Sections 2.5.2 and 2.5.3.

statistics of θ have to be modeled appropriately. Though seemingly simple, this problem has earned the attention of many researchers in the past, and continues to remain a challenging one especially when the scaling intensities are far away from unity.

(M2) **Amplitude modulation (AM):** The factor graph corresponding to this model is shown in Figures 2.5 and 2.6. This can be thought of as a generalization of (M1) where the scaling parameter varies spatially. Let $\mathscr{I}_{\mathscr{P}} \triangleq \{\mathscr{I}^{(1)}, \mathscr{I}^{(2)}, \ldots \mathscr{I}^{(J)}\}$ be a partition (e.g., rectangular tiling) of the pixel domain $\mathscr{I}$. The AM field is parameterized by a collection of parameters θ, and we model this as a Gaussian Markov random field with unit (or in general, some known) mean and first-order neighborhoods, acting independently across *nonoverlapping* subblocks of the host. The dependencies within each subblock are described by

$$p(\theta^{(j)}) = \frac{1}{Z_2} \prod_{i \in \mathscr{I}^{(j)}} \psi(\theta_i - \mu_\theta, \sigma^2_{\theta a}) \psi(\theta_i - \theta_{iE}, 4\sigma^2_{\theta b}) \psi(\theta_i - \theta_{iN}, 4\sigma^2_{\theta b}), \tag{2.13}$$

$\forall j = 1$ to J, where $\theta^{(j)}$ is the amplitude modulation field acting on $\mathscr{I}^{(j)}$, $\mu_\theta = 1$, and all other quantities, including the (blockwise) cycling around effect, are defined similarly as in (2.12). For example, the constant scaling operation on the entire image (as in (M1)) corresponds to the simple tiling $\mathscr{I}_{\mathscr{P}} = \{\mathscr{I}\}$ and the limiting case of $\sigma_{\theta b} \downarrow 0$. Any other arbitrary tiling with $\sigma_{\theta b} \downarrow 0$ will correspond to a piecewise constant AM distortion. Introducing additional dependencies in the AM field across subblocks can only make the inference problem easier (but possibly with an added computational cost) as this effectively decreases the number of independent parameters to estimate. Similarly, decreasing the information embedding rate can also only make the inference task easier.

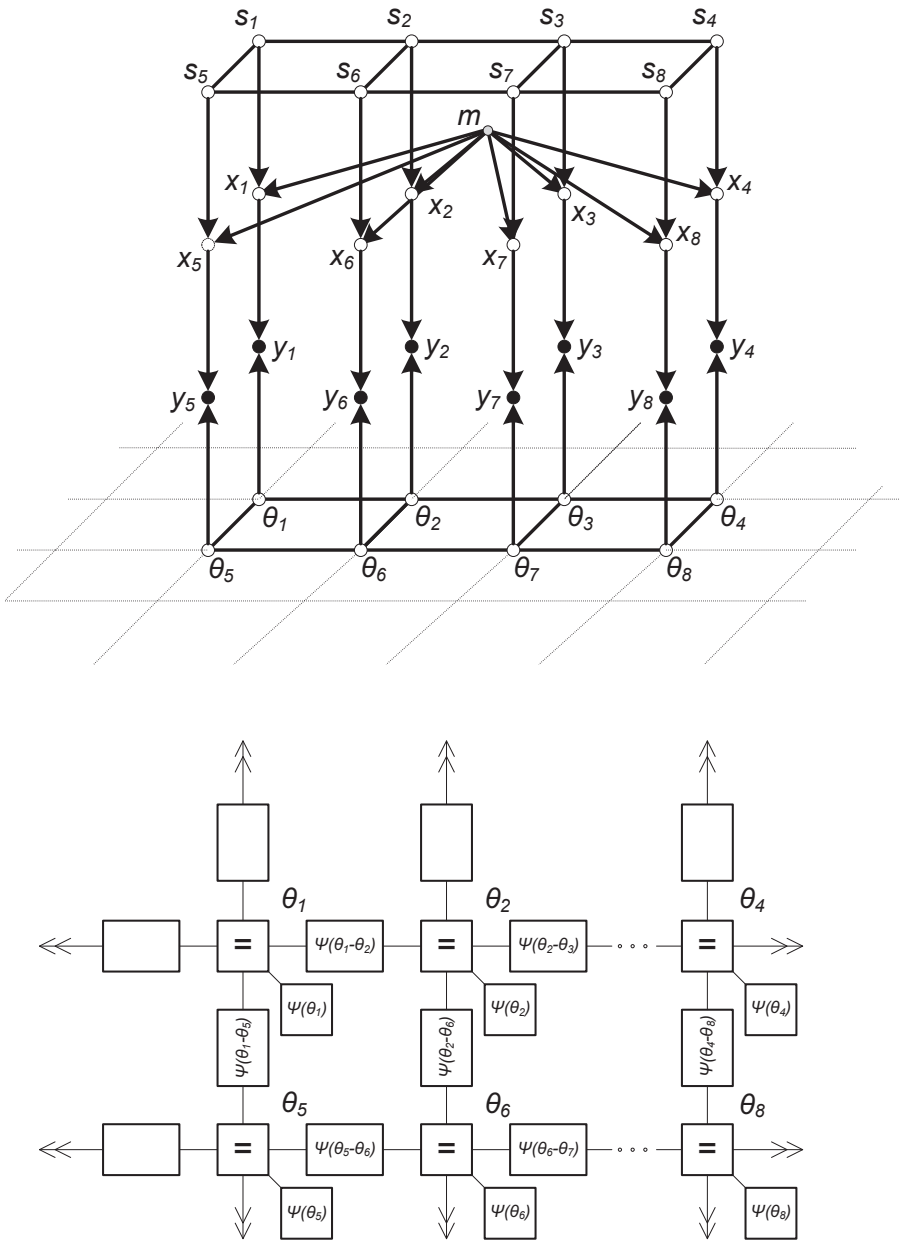


Figure 2.5: **Top:** A dependency graph for the full probabilistic model corresponding to (M2), Section 2.5. For clarity, we only show a 4×2 cross-section of the graph into which one bit is embedded. The embedding rate is $R = 1/8$. **Bottom:** Factor graph (Forney-style) corresponding to the 'θ-plane' in the dependency graph. A few of the factor nodes are labeled for the sake of exposition. Due to space constraints, we omit the second argument to $\psi(\cdot)$ in the diagram. For added clarity, we also show the full factor graph skeleton for the AM case in Figure 2.6.

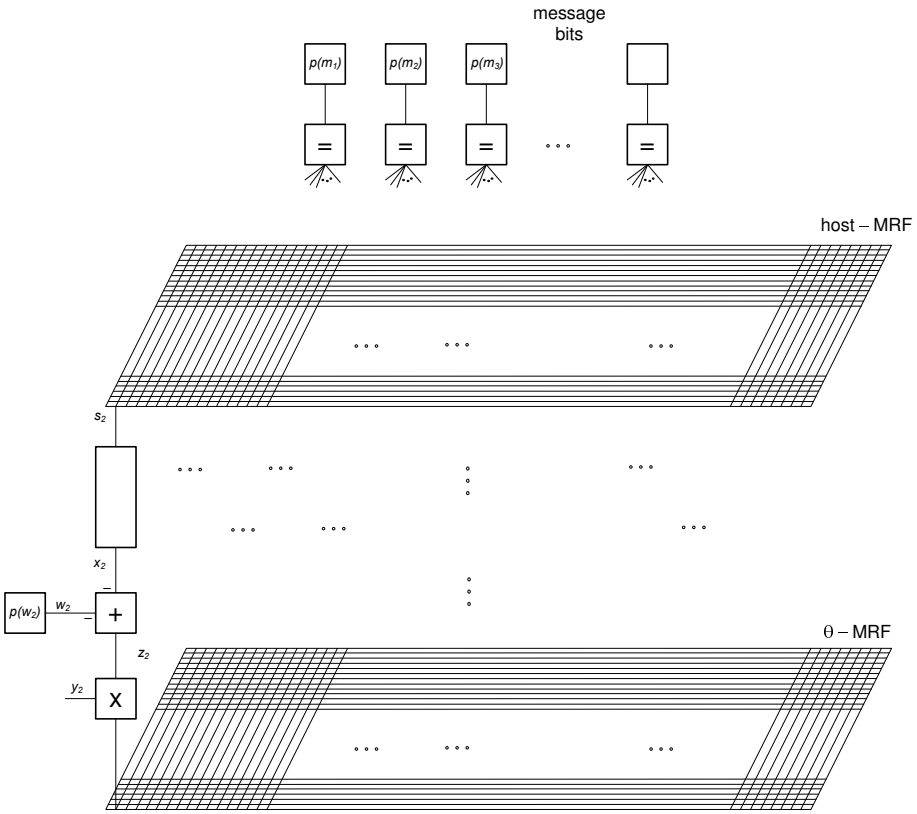


Figure 2.6: A skeletal representation of the Forney factor graph corresponding to (M2), amplitude modulation. The meshes labeled "host-MRF" and "θ-MRF" are constructed in a manner similar to the bottom graph of Figure 2.5 and are not shown in detail due to space constraints. Vertical branches, similar to the one depicted above, exist between all corresponding (vertically aligned) nodes on the "host-MRF" and "θ-MRF" meshes. Again, we omit drawing those to reduce clutter.

In this work, we will assume that the AM field acts independently on 8×8 blocks of the host. In the limiting case of $\sigma_{\theta b} \downarrow 0$, this constitutes an AM distortion with 32×32 independent parameters. Though this may not constitute a typical "smooth" distortion operation, we report results for this case to illustrate the power of our approach. The same algorithm can however be applied to smoother (and typically more commonly encountered) distortions, as will be soon seen in Section 2.5.2.

(M3) **LSI filtering:** $\mathbf{y} = \mathbf{z} \star \mathbf{h}_\theta$, where $\star$ denotes linear convolution and $\mathbf{h}_\theta$ is an arbitrary two-dimensional linear shift-invariant (LSI) filter parameterized by a random variable θ whose statistics are assumed to be known to the receiver. For LSI filtering operations, we perform the watermark embedding in the Fourier domain, while ensuring that the marked signal remains real valued. The choice of Fourier domain is motivated by the desire to induce a factor graph that has as few loops as possible (also see comments in Section 2.6). This embedding procedure is explained in detail in [34] (Algorithm 1). For illustration purposes, in this work, we will use the filter $\mathbf{h}_\theta$ obtained by cascading the zero-phase 'exponential' one-dimensional (low-pass) filter applied in both directions:

$$\mathbf{g}_\theta = \text{IDFT}\big[\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \dots \theta^{(2)}, \theta^{(1)}\big], \tag{2.14}$$

where[6] $\theta^{(i)} \triangleq \max(\theta^i, \epsilon)$, $\theta \sim \mathcal{N}(0.5, 0.1^2)$, and IDFT denotes the inverse DFT. The constant ϵ is small, say 10^{-4}, and is employed to prevent numerical instabilities during channel inversion.

(M4) **Fractional (spatial) shift:** This can be seen as a special case of (M3), but as in many other watermarking papers, we shall give it some special attention. In the one-dimensional case, $y(n) = z(n-\theta)$ for integer shift θ. If θ is not an integer,

$$y(n) = \sum_i h_i(\theta) z(n-i) \tag{2.15}$$

is a resampled version of the shifted, interpolated signal $\mathbf{z}$, where

[6] Note that i is merely a superscript index in $\theta^{(i)}$, but an exponent in θ^i.

$h_i(\theta)$ are the taps of the interpolation filter (would be a sinc for bandlimited interpolation; we will use the bicubic kernel). If θ is a constant, (2.15) is a particular linear shift-invariant filter. If θ varies slowly over space (as is the case with spatial warping), (2.15) is a linear shift-variant filter. This model can be extended to images by applying the same one-dimensional filter along each direction.

(M5) **Blockwise filtering and shift:** The filter is assumed to act independently on 8×8 blocks of the host; this means a total of 32^2 different (and possibly correlated) values of θ for a 256×256 host. We use the same models for LSI filtering and shift as in (M3) and (M4), respectively, the only difference being that the filter parameter θ now varies spatially. The 32×32 θ field is modeled to be drawn from a Gaussian Markov random field as in (M2) with parameters $\tilde{\mu}_\theta$, $\tilde{\sigma}_{\theta a}$ and $\tilde{\sigma}_{\theta b}$.

Joint estimation of the distortion parameters and the embedded bits is performed *independently* on nonoverlapping 8×8 subblocks of the host, and this is found to be sufficient to yield good decoding performance. Yet, this is a suboptimal approach because the observed image samples are correlated due to the dependencies introduced by the host. Using more samples (e.g., the entire host image) for performing the joint estimation could improve decoding performance, but at the significant cost of complexity and speed.

2.5.1 Synthetic Host Signals

Here, we present some experimental results on synthetic host signals. A 256×256 host with $\mu_s = 0$ and $\sigma_{sa} = \sigma_{sb} = 50$ is drawn according to (2.12) by Gibbs sampling [47]. Watermarking is done as described earlier for various distortions, (M1) through (M5). The distortion compensation parameter is chosen in agreement with Eggers' value, i.e., $\alpha = \sqrt{\frac{WNR}{WNR+2.7}}$, which is nearly optimal in the sense of maximizing the rate of reliable communication over an additive white Gaussian noise channel [26]. The host-to-watermark ratio (HWR, see (2.6)) is 25 dB, and the watermark-to-noise ratio ($WNR \triangleq D_1/D_2$) is varied from -2 dB to 1 dB. Here, D_2 denotes the variance of additive Gaussian noise $\mathbf{w}$. Independent bits are embedded in 4×2 blocks of the host. As explained earlier, the same bit is embedded in each component (pixel) of the subblock, thereby inducing a simple rate $1/8$ repetition error correction code within each 4×2 subblock of the host. We evaluate the bit error rate

$$P_e = \frac{1}{n_b} \sum_{i=1}^{n_b} \mathbb{1}_{\{m_i \neq \hat{m}_i\}} \tag{2.16}$$

for the estimator-decoder described in Section 2.4, and compare it with that of the coherent decoder (denoted by P_e^*) that knows θ. Numerical results for various values of model parameters, and for each of the distortion models (M1) through (M5) are given in Figures 2.7, 2.8, 2.9, 2.10 and Tables 2.2, 2.3.

The performance of the noncoherent decoder is seen to be *almost as good* as that of the coherent one in most experiments, for a wide range of distortion intensities, 'mild' to 'strong'. To the best of our knowledge, no previous work has demonstrated such resilience to distortions with such strong intensities. In [10], where Balado *et al.* report results for scaling and fractional shift distortions, the decoder's performance begins to deteriorate rapidly outside a narrow scaling range of (0.9,1.1).

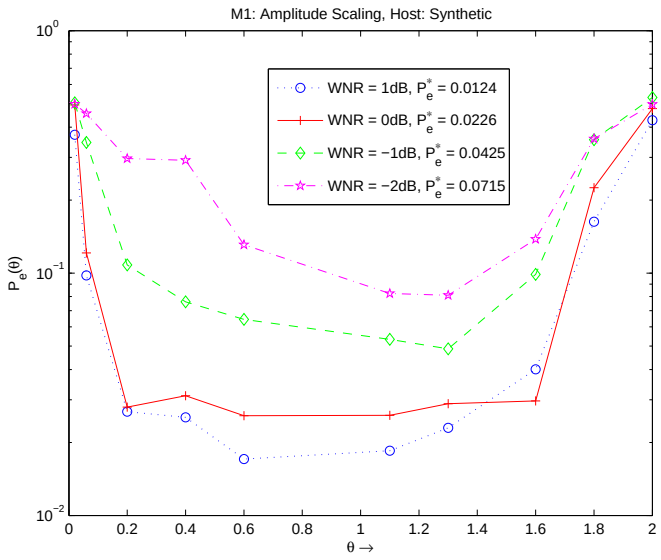


Figure 2.7: Numerical results for (M1) – Amplitude scaling, with synthetic host, and scaling parameter θ. $HWR = 20\,\text{dB}$.

Table 2.2: Numerical results for (M2) – Amplitude modulation by Gauss-Markov amplitude field θ. $HWR = 25\,\text{dB}, WNR = 0\,\text{dB}$.

#	$\sigma_{\theta a}$	$\sigma_{\theta b}$	P_e (M2)	
			Section V-A	Section V-B
1.	1	10^{-4}	0.027	0.035
2.	5	10^{-4}	0.060	0.0598
3.	0.1	0.1	0.028	0.041
4.	0.1	0.5	0.031	0.048
P_e^*			**0.0253**	**0.0310**

A similar behavior can be seen in the results of Miller *et al.* [19] for scaling intensities above 1.1. In contrast, our decoders nearly match the performance of the coherent decoder even when the scaling intensities are far away from unity. Similarly, for the fractional shift distortion analyzed by Balado *et al.* [10], the error probability numbers quickly rise up to 0.5 for shifts as low as 0.2; in contrast, the graph-based decoder is seen to perform well for shifts even as high as 2.5. Similar comparisons hold for other types of distortions as well.

Finally, it is also illustrative to compare the decoding performance of

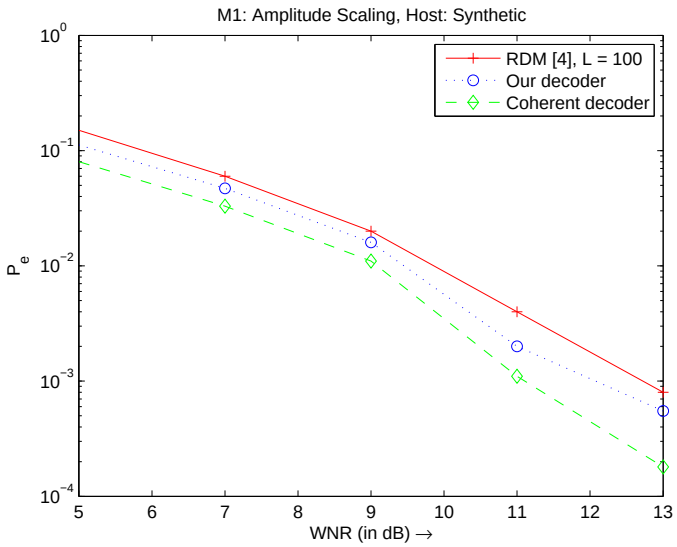


Figure 2.8: Numerical results for (M1) – Comparison of our decoder with that of rational dither modulation (RDM). $HWR = 25\,\text{dB}, R = 1$ bit/sample. L denotes the 'memory length' in RDM.

Table 2.3: Numerical results for (M5) – Blockwise filtering (Rows 1, 2, 3, 4), Blockwise shift (Rows 5, 6, 7, 8). $HWR = 25\,\text{dB}, WNR = 0\,\text{dB}$.

#	$\tilde{\mu}_\theta$	$\tilde{\sigma}_{\theta a}$	$\tilde{\sigma}_{\theta b}$	P_e (M5)	
				Section V-A	Section V-B
1.	0.4	0.1	0.1	0.0331	0.0399
2.	0.4	0.2	0.1	0.0318	0.0364
3.	0.6	0.05	0.01	0.0292	0.0602
4.	0.6	0.01	0.01	0.0377	0.0591
5.	1	0.5	0.1	0.0411	0.0551
6.	1	0.5	0.1	0.0597	0.0347
7.	1.5	0.05	0.01	0.0331	0.0529
8.	1.5	0.01	0.01	0.0318	0.0499
P_e^*				**0.0253**	**0.0310**

our scheme with that of 'rational dither modulation' or RDM [13], a recently proposed modification of the standard QIM encoder to specifically combat scaling distortions. Figure 2.8 plots the performance of our scheme against that of the coherent decoder, and the encoder-decoder setup of RDM. In the plots, L denotes the memory length parameter in RDM; more details can be found in [13]. Simulations

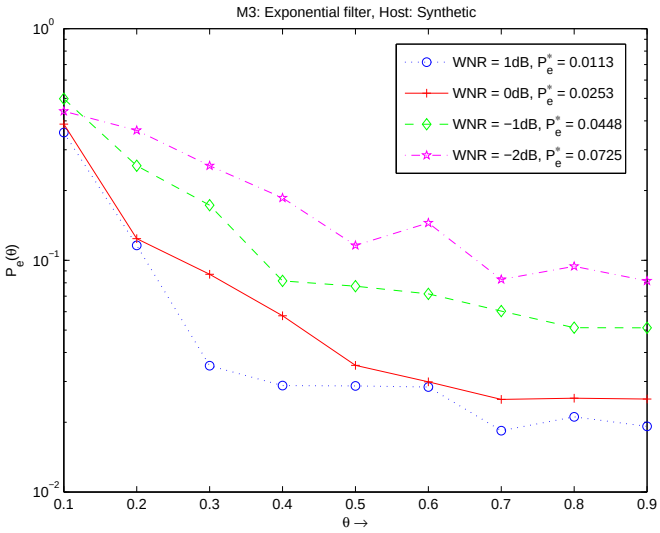


Figure 2.9: Numerical results for (M3) – LSI filtering with an 'exponential' low-pass filter, and synthetic host. $HWR = 25\,\text{dB}$.

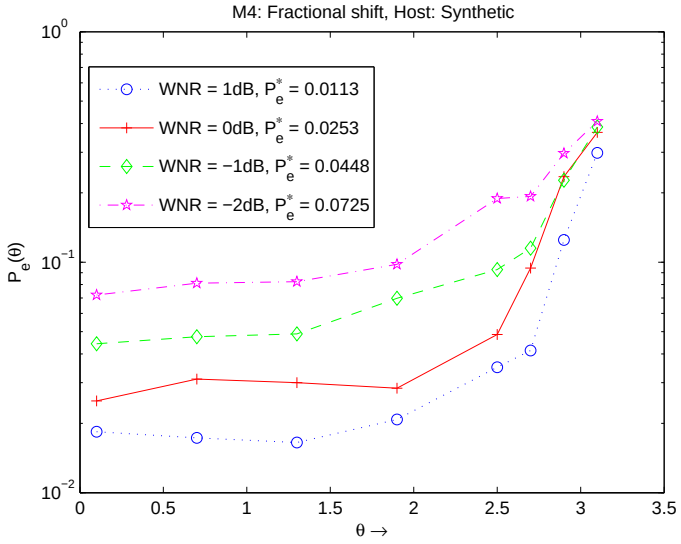


Figure 2.10: Numerical results for (M4) – Fractional shift by θ, with synthetic host. $HWR = 25\,\text{dB}$.

are performed on a one-dimensional host consisting of 256 samples, drawn from a Gaussian MRF with parameters $\mu_s = 0, \sigma_{sa} = 50, \sigma_{sb} = 50, HWR = 25\,\text{dB}, WNR \in [5\,\text{dB}, 15\,\text{dB}]$. The embedding rate is one bit per sample, i.e., no error correction code is employed. As one can note from Figure 2.8, our system outperforms RDM uniformly over this

range of WNRs, which is encouraging considering that our encoder is generic and was not specifically tailored to handle scaling. Also noteworthy is that while RDM is designed to be asymptotically invariant to constant scaling distortions, the authors also report that their method does not satisfactorily combat spatially varying scaling. This is disappointing, as spatially varying scaling can be thought of as the most elementary generalization of spatially constant scaling distortion, and yet the benefits of RDM do not carry over. In contrast, we are able to report good distortion-resilience even on spatially varying scalings of the watermarked image, by using the same, unified decoder framework.

2.5.2 Real Host Images

The experiments of Section 2.5.1 were repeated on a 256×256 image of Lena. We perform watermarking, as described in earlier sections, but in the wavelet domain. We use the difference between the original image and the low-pass approximation version obtained from the 16×16 approximation coefficients of a four-level Daubechies-4 discrete wavelet transform (DWT) as input to marking algorithms described in earlier sections. As before, the host-to-watermark ratio is maintained at 25 dB, and the watermark-to-noise ratio is varied from -2 dB to 1 dB. The parameters σ_{sa} and σ_{sb} for modeling the difference image as in (2.12) are obtained as pseudo-maximum likelihood estimates (see Appendix A), and shared with the decoder. As we will see later in Section 2.5.3, accurate estimates of these parameters are not really necessary. Numerical results for various values of model parameters, and for each of the distortion models (M1) through (M5), are given in Figures 2.11, 2.12, 2.13 and Tables 2.2, 2.3.

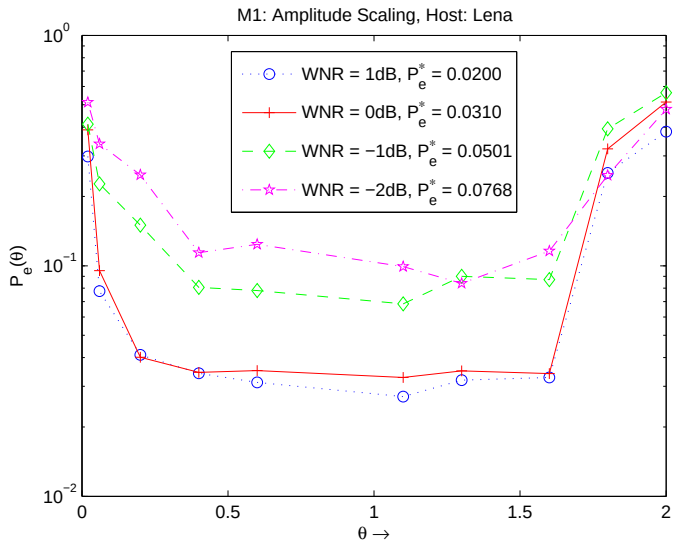


Figure 2.11: Numerical results for (M1) – Amplitude scaling, with Lena, and scaling parameter θ. $HWR = 25\,\text{dB}$.

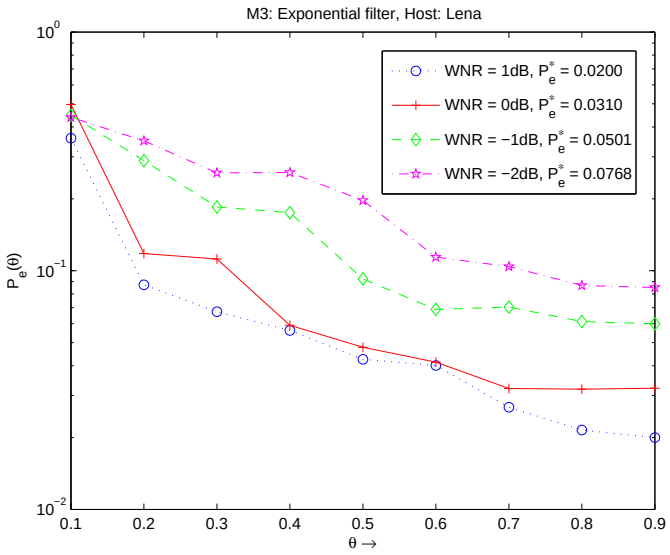


Figure 2.12: Numerical results for (M3) – LSI filtering with an 'exponential' low-pass filter, and Lena. $HWR = 25\,\text{dB}$.

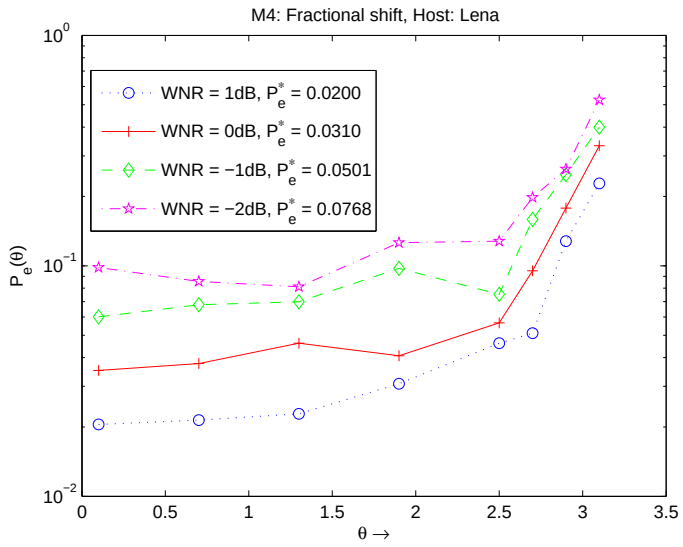


Figure 2.13: Numerical results for (M4) – Fractional shift by θ, with Lena. $HWR = 25$ dB.

The images of Figure 2.14 and 2.15 illustrate the intensity of various distortions. As mentioned earlier, all the numbers reported in Table 2.2 correspond to an MRF acting independently on nonoverlapping 8×8 subblocks of Lena (Figure 2.14(b)). However, primarily to serve as a visual aid, we also show images corresponding to a 64×64 tiling as well (Figure 2.14(c)). It may be noted that the latter is in fact an easier inference problem, but our algorithm is capable of producing near-coherent decoding even on the former. Also shown (in Figure 2.14(d)) is an example of a "smooth" AM distortion, for which the same algorithm can be applied blindly, so long as the AM field stays roughly a constant over 8×8 subblocks of the image (as is the case in Figure 2.14(d)).

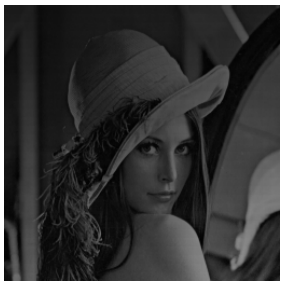
2.5.3 Robustness to Host Modeling Mismatch

Here, we investigate the robustness of our algorithm to mismatches in host modeling. This is an important issue, especially when working with natural images, as it may not be feasible (or practical) to obtain

(a) Original Lena

(b) For Table 2.2, Row 2

(c) AM distortion #2



(d) AM distortion #3

Figure 2.14: Various desynchronized versions of Lena. It may be noted that no noise has been added to the images above (i.e., $\mathbf{w} = 0$). Also see Figure 2.15.

an accurate characterization of the underlying statistics of the host image.

We repeat the experiments of Section 2.5.1, using the AM model from (M2), but with a twist. Host samples are drawn from a mismatched model of (2.12), with σ_{sa}, σ_{sb} replaced by $\tilde{\sigma}_{sa}, \tilde{\sigma}_{sb}$ respectively. The performance of the mismatched decoder is reported in Table 2.4. The results show that the decoder is capable of tolerating host modeling mismatches to a significant extent, a property that will be useful when working with natural and photographic images as it may not be possible to get accurate modeling parameter estimates for the same.

(a) For M3, $\theta = 0.7$

(b) For M3, $\theta = 0.5$

(c) For M3, $\theta = 0.3$

(d) Blockwise filtering

Figure 2.15: More desynchronized versions of Lena. As in Figure 2.14, no noise has been added to the images above (i.e., $\mathbf{w} = 0$).

Table 2.4: Numerical results for Section 2.5.3 using the AM distortion model of (M2). $P_e^* = 0.0253$, for a correctly matched decoder, and $HWR = 25\,\text{dB}, WNR = 0\,\text{dB}$. Here, $\gamma = \tilde{\sigma}_{sa}/\sigma_{sa} = \tilde{\sigma}_{sb}/\sigma_{sb}$ captures the extent of model mismatch.

#	$\sigma_{\theta a}$	$\sigma_{\theta b}$	γ	P_e (M2)
1.	5	10^{-4}	1	0.060
2.	5	10^{-4}	0.5	0.0591
3.	5	10^{-4}	2	0.0678
4.	0.1	0.1	1	0.028
5.	0.1	0.1	0.5	0.041
6.	0.1	0.1	2	0.050

We conclude this section with a note on running times for our decoding algorithm. All experiments were performed on a Windows machine with 32 bit OS, 2.67 GHz quad-core processors and 4GB RAM. The code was not fine-tuned for parallel computation and takes anywhere between 10 to 20 minutes to decode a rate 1/8 message (watermark) from a 256×256 host, with $\tau = 4$.

2.6 Discussion and Caveats

First, as seen in the previous section, one of the most striking positives of our approach is the significantly increased resilience to various distortions. Explicit comparisons with results from [10] have revealed increased resilience ranges of up to 10× in the case of scaling and even more for pure shift distortions. Another important power of graph-based decoders is their ability to handle high-dimensional distortion parameterizations. For example, in the AM experiments (M2), we saw that noncoherent decoding was feasible even when faced with the challenging task of estimating 32^2 independent distortion parameters. This is important for two reasons: (1) Such a decoding approach is significantly more efficient (w.r.t. algorithm execution time) than a brute force search [34]. (2) This property sets it apart from the other decoders proposed to date; for example, the decoder of [10] uses 15×10^3 host samples to estimate a single unknown parameter. Thus, the 'divide and conquer' iterative estimation approach has its clear benefits.

Secondly, we saw in Section 2.5.3 that decoding performance is good even if the receiver does not know exact host signal statistics; in other words, crude host modeling will suffice and does not influence the decoding by much. However, the same is not necessarily true of distortion modeling. In all of our reported experiments, we have assumed that the receiver is fully cognizant of distortion parameter statistics. If this was not the case, the receiver would face a problem in the sense that

the lack of prior information about θ could result in increased time to convergence. However, so long as we have a sufficiently large number of observed signal samples, it is not critical that the statistical modeling of θ should be exact. For instance, in (M1) where θ is the scaling parameter, we still obtain good results if θ was uniformly distributed over $(0, 2)$, but experiments modeled it to be drawn from a Gaussian distribution, say $\mathcal{N}(1, 0.3^2)$. This is not surprising, as the influence of the prior vanishes with increasing number of observed samples in any estimation problem (subject to regularity conditions). To summarize, exact knowledge of parameters describing the statistics of θ is not absolutely necessary, but we have observed it does help in terms of convergence speed, especially when the number of observed signal samples per number of distortion channel parameters is moderate.

Another implicit assumption made during the construction of factor graphs corresponding to our system is that a message bit has been emdedded in each subblock of the host. An interesting question to ask would be, "What if this is not the case?" In this case, the decoding rules at the decoder can be modified a little so we have *three* possible outputs corresponding to each message bit: 0, 1 or ϕ. Typically, such a decoding rule would take the form:

$$\hat{m}_i = \begin{cases} 1 & \text{if } \frac{\mu_{m_i}(1)}{\mu_{m_i}(0)} > 1 + \eta_1 \\ 0 & \text{if } \frac{\mu_{m_i}(1)}{\mu_{m_i}(0)} < 1 - \eta_2 \\ \phi & \text{otherwise,} \end{cases}$$

where η_1 and η_2 are positive constants, and μ_{m_i} is the message corresponding to the i^{th} information bit. This is not a heuristic, but in fact follows (in spirit) from the idea behind soft versus hard decoding. In our simulations, we pick $\eta_1 = \eta_2 = 0$, which is optimal when the decoder knows that $\mathbf{m} = \phi$ is an impossible event (this is indeed the case with the model described in Section 2.2).

And finally, an important caveat – tractable factors and graph loops. As is already evident, loops are unavoidable in (almost) all the systems we consider. However, care needs to be taken to avoid loops wherever possible, as such intelligent graph constructions do seem to play a vital role in influencing the convergence properties of the message-passing algorithm. While loops are best avoided wherever possible, in general, graphs with longer loops seem to offer a more conducive platform for convergence as opposed to one with many short loops. Care also needs to be taken to design graphs that avoid messy factors, wherever possible. As an example, one can equivalently implement the LSI filter of (M3) in the spatial domain instead of the frequency domain. However, the former approach yields factors that are much more complicated than what we saw in the latter approach, and this might lead to computationally intensive message update steps. Further, as illustrated in Figure 2.16, the spatial domain treatment of the problem will result in a graph that has many more loops than is necessary (the number of loops scales steeply with the number of nonzero filter taps), and based on our experiments, convergence does not happen in a reasonable amount of time (in fact, we cannot guarantee that the messages will eventually stabilize). Perhaps better convergence could be obtained if we chose to perform the belief propagation updates differently instead of discretized messages; for example, nonparametric message-passing could be used. We have not explored such options in this thesis.

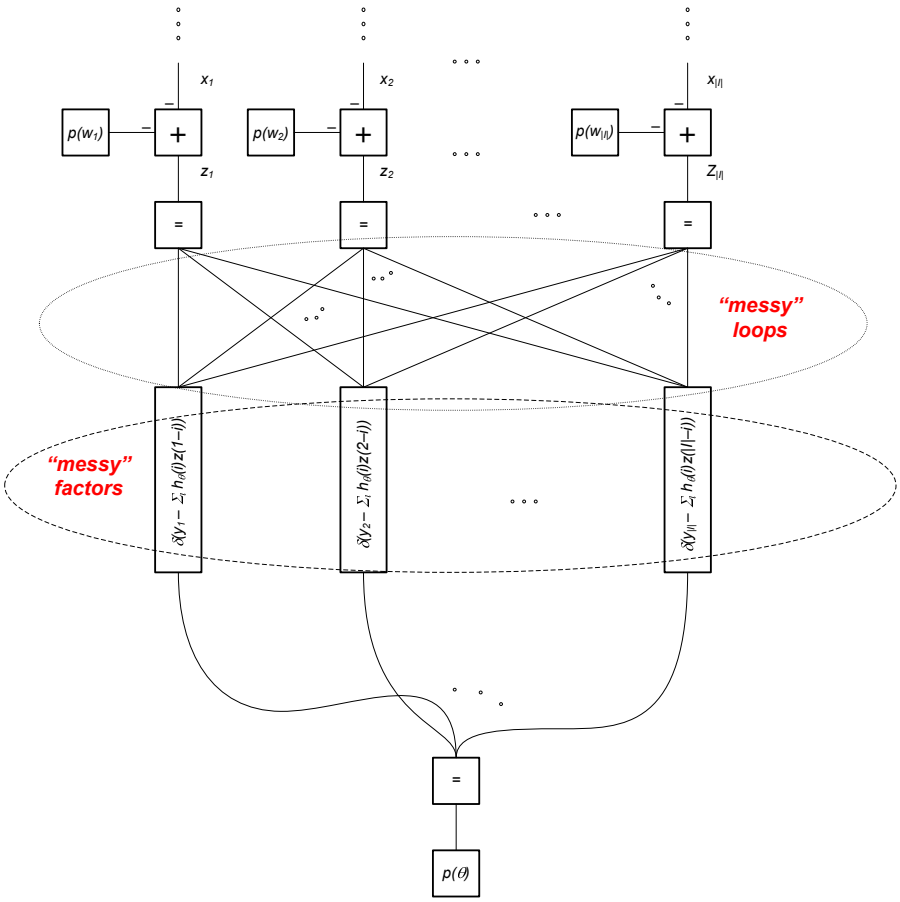


Figure 2.16: A spatial domain approach to handling distortions of type (M3). For simplicity, we only illustrate the 1d case here. Messy factors, those with degree more than 3 or 4, significantly increase the computational complexity of discretized messages' updates.

Chapter 3

Universal Divergence-Rate Estimators for Steganalysis in Timing Channels

This chapter is organized as follows. Following some notation and definitions in Section 3.1, we define our system model in Section 3.2 and introduce queue-based codes in Section 3.3. Section 3.4 introduces the role of the Kullback-Leibler (KL) divergence-rate in the context of covert communication over timing channels. Section 3.5 provides a survey of results from the literature pertaining to divergence-rate estimation, one of which we extend for use in our problem setting through a conjecture in Section 3.6. Our proposed divergence-rate estimator is given in Section 3.7, and we give some experimental results in Section 3.8.

3.1 Notation

The following notation will remain in effect throughout this chapter:

- Random variables and their realizations are denoted by uppercase and lowercase letters, respectively. Boldface letters denote sequences of natural numbers.
- $\mathbf{x}_i^j$ denotes the subsequence $(x_i, \dots x_j)$; $\mathbf{x}^i = \mathbf{x}_1^i$; and $\mathbf{x} = \mathbf{x}_{-\infty}^{\infty}$.
- The probability mass function of $\mathbf{X}^n$ is denoted by $p_{\mathbf{X}^n}$, and the pdf evaluated at an instance $\mathbf{x}^n$ is $p_{\mathbf{X}^n}(\mathbf{x}^n)$.
- "log" denotes the base 2 logarithm, and "ln" denotes the natural logarithm.

Definitions: Let $\mathbf{X} = \{X_n\}_{n\mathbb{Z}}$ denote a stationary process taking values from a finite or countable alphabet, and underlying distribution P; and $\mathbf{Y} = \{Y_n\}_{n\mathbb{Z}}$ denote a stationary process taking values from a finite or countable alphabet, and underlying distribution Q. Some definitions follow:

- $H(X) \triangleq -\sum_x p(x) \log p(x)$ denotes the entropy (in bits) of the discrete random variable X, distributed according to the probability mass function p.
- $D(p||q) \triangleq \sum_{x\in\mathbb{N}} p(x) \log \frac{p(x)}{q(x)}$ denotes the KL divergence between two pmf's p and q.
- $H'(P)$ denotes the entropy-rate (in bits) of the process $\mathbf{X}$ with distribution P, and is defined by

  $$H'(P) = \lim_{n\to\infty} \frac{1}{n} H(X_1^n).$$

  If $\mathbf{X}$ is stationary, the above is equivalent to

  $$H(P) = \lim_{n\to\infty} \mathbb{E}[-\log P(X_0|X_{-n}^{-1})].$$

- $D'(P\|Q)$ denotes the divergence-rate (in bits) between probability laws P and Q, and is defined by

  $$D' = \lim_n \frac{1}{n} D(p_{\mathbf{A}^n} \| p_{\mathbf{D}^n}),$$

  assuming the limit exists.

3.2 System Model

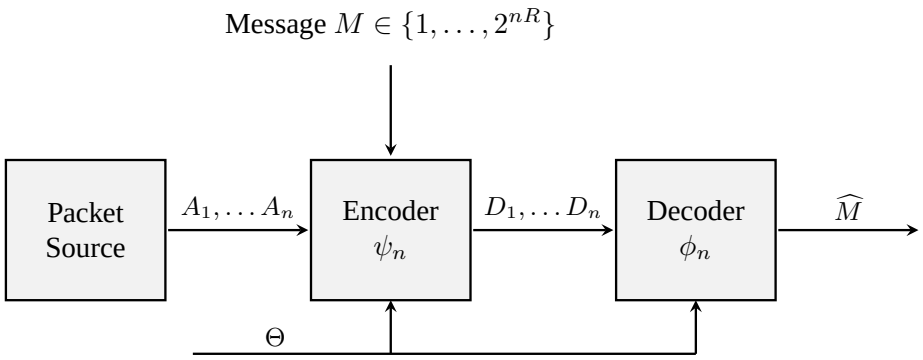


Figure 3.1: System model for timing channels

The system model for timing channels is shown in Figure 3.1. The covertext is modeled as a sequence of interarrival times $\mathbf{A}^n = (A_1, A_2, \ldots A_n)$ of i.i.d. samples drawn from a pmf $\{p_A(a), a \in \mathbb{N}\}$. A message M uniformly distributed over $\mathscr{M} = \{1, 2, \ldots 2^{nR}\}$ is to be embedded in $\mathbf{A}^n$ and transmitted to a decoder. The stegocoder produces a stegotext $\mathbf{D}^n$ through a function $\psi_n(\mathbf{A}^n, M)$ in order to convey the message M to the decoder reliably. The covertext and stegotext are required to be close according to some distortion metric, a popular choice motivated by information-theoretic justifications being the KL divergence-rate between the covertext and stegotext processes

$$D' = \lim_n \frac{1}{n} D(p_{\mathbf{A}^n} \| p_{\mathbf{D}^n}), \tag{3.1}$$

assuming the limit exists [48,49]. The decoder does not have access to the original covertext $\mathbf{A}^n$ and produces an estimate $\widehat{M} = \phi_n(\mathbf{D}^n) \in \mathscr{M}$. The code (ψ_n, ϕ_n) is randomized using a random variable Θ known only to the stegocoder and decoder. The expected latency introduced by the code after transmission of n packets is $\tau_n = \mathbb{E}\sum_{j=1}^{n}(D_j - A_j)$ where the expectation is taken with respect to M and Θ.

As alluded to in Section 1.2.2, stegocodes with high latency render

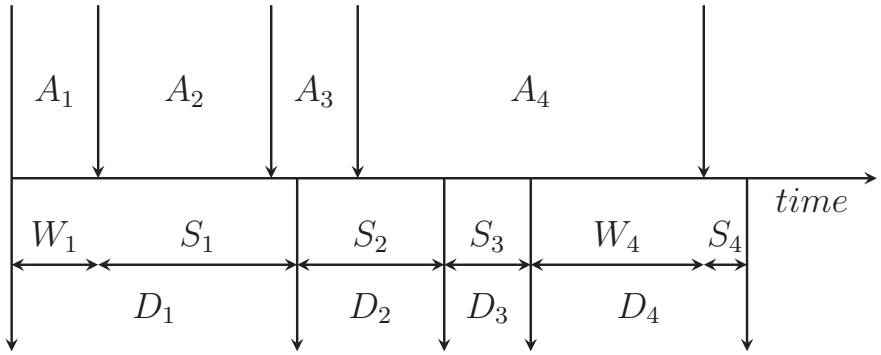


Figure 3.2: Interarrival (**A**), idle (**W**), service (**S**) and interdeparture (**D**) times for a queue.

the covert communication channel detectable with even elementary statistical operations. In this thesis, we aim to devise steganalysis tools that will work well for much 'smarter,' e.g., queue-based codes.

3.3 Queue-Based Timing Channel Stegocodes

Figure 3.2 is a pictorial representation of a queue, which is basically a nonlinear system with memory. The basic mathematical model for a (single server) queue is as follows. Packets randomly spaced in time arrive at the queue's input expecting service. The interarrival times between packets are denoted by $\mathbf{A} \triangleq \{A_i\}_{i\in\mathbb{N}}$. Packets receive service instantaneously if the queue server is idle. If not, they are subject to a positive service time. The service time corresponding to the i^{th} packet is S_i and $\mathbf{S} \triangleq \{S_i\}_{i\in\mathbb{N}}$. W_i denotes the time spent by the queue idling, before servicing the i^{th} packet, and after the departure of the $(i-1)^{th}$ packet; $\mathbf{W} \triangleq \{W_i\}_{i\in\mathbb{N}}$. $\mathbf{D} \triangleq \{D_i\}_{i\in\mathbb{N}}$ denotes the interdeparture times. It is possible to compactly capture the above model through the famous

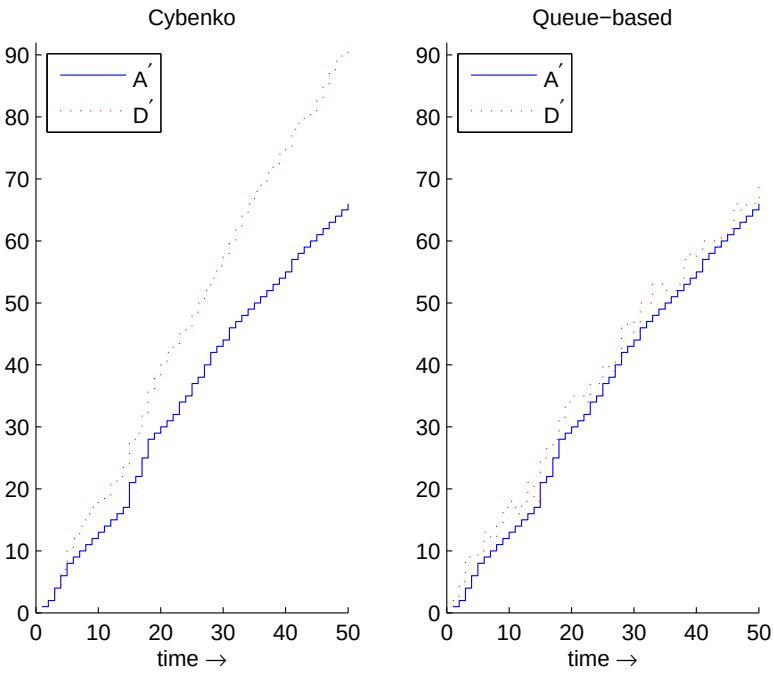


Figure 3.3: The arrival process $A'_n = \sum_{i=1}^{n} A_i$ and the departure process $D'_n = \sum_{i=1}^{n} D_i$ for the code of Cybenko *et al.* (also see Section 3.8.1), left, and the simple queue-based code of (3.4), right.

Lindley's equations [50]:

$$D_i = W_i + S_i, \tag{3.2}$$

$$W_i = \left| \sum_{j=1}^{i} A_j - \sum_{j=1}^{i-1} D_j \right|^{+}, \quad i \in \mathbb{N}. \tag{3.3}$$

Queue-based stegocodes were introduced in [48] to circumvent a major limitation – the 'drift problem' – that cripples several encoding functions proposed in the literature (e.g., [37, 38, 51, 52]). The drift problem (Figure 3.3) causes the latency of the stegocode to linearly increase with time, thereby forcing the stegocoder to actively transmit only intermittently, in order to keep the overall delay small. This results in an inefficient exploitation of the covert channel, and makes the communication easily detectable.

Queue-based codes put forth a new approach to designing the encoding function wherein the information bits are embedded in the covertext $\mathbf{A}^n$ via a queue action. For example, assume that p_A is the geometric distribution with parameter λ, i.e., the interarrival process $\mathbf{A}^n$ is the discrete-time version of a Poisson process. Consider the simple code that modifies the i^{th} idle time W_i (corresponding to the i^{th} packet) of the queue into an interdeparture time D_i to embed a binary bit $b_i \in \{0,1\}$ as follows:

$$D_i = \begin{cases} W_i + S_i' + 1, & W_i + M_i \quad \text{even} \\ W_i + S_i', & W_i + M_i \quad \text{odd} \end{cases} \tag{3.4}$$

where S_i' is an odd random number drawn from the following pmf:

$$p_{S_i'}(s_i') = \mu(2-\mu)(1-\mu)^{s_i'-1} \quad s_i' = 1,3,5,\ldots \tag{3.5}$$

This code does not suffer from the drift effect, as illustrated in Figure 3.3. Also, evidently, reliable decoding is possible via simple modulo 2 operations on the interdeparture times – an even interdeparture time conveys an information bit zero, while an odd interdeparture time conveys a one.[1] It seems clear from Figure 3.3 that the queue-based code of (3.4) is 'less' detectable than that of Cybenko's [38]. In this work, we seek new tools that will help us establish such claims formally.

3.4 Kullback-Leibler Divergence-Rate

The Kullback-Leibler (KL) divergence-rate, defined in Section 3.2, is used to measure the extent of dissimilarities between two stochastic processes, and plays a pivotal role in both the design and steganalysis

[1] Note however that the code of (3.4) is not perfectly secure as the statistics of the arrival and departure process do not match exactly, resulting in a nonzero divergence-rate between the two. The problem of constructing high rate, perfectly secure codes that can be decoded reliably is explored in [49].

of timing channels. From a design point of view, the divergence-rate is an integral part of the definition of secure stegocodes, and hence is an important optimization criterion used in their design [48, 49]. From a steganalysis viewpoint, the divergence-rate manifests itself in the form of goodness-of-fit tests [53–55] that are powerful statistical procedures used to decide whether a particular sequence of observations came from a certain probability distribution or not. For example, in the absence of any covert operations, the statistical model for interdeparture times should coincide with that of the interarrival times. Given traffic data (interdeparture times) $d_1, d_2, \ldots d_n$, the following test declares the data as being stegodata if the estimated divergence-rate $\hat{D}'$ exceeds a prescribed threshold τ:

$$\hat{D}' > \tau, \tag{3.6}$$

where $\hat{D}'$ is an *estimate* of D', the KL divergence-rate (3.1) between the interarrival process $\mathbf{A}^n$ and observed interdeparture processes $\mathbf{D}^n$. Here, $\hat{D}'$ is estimated from the observed interdeparture times $d_1, d_2, \ldots d_n$, and *given* reference interarrival times $a_1, a_2, \ldots a_n$. The performance of (3.6) can then be measured by the receiver operating characteristic (ROC) curve which quantifies the tradeoff (parameterized by τ) between false positives and negatives.

3.4.1 Optimal Test?

Here, we briefly discuss the rationale behind choosing the test of (3.6) for our setup. One of the early works to study the above test was Hoeffding in [56], where he restricted attention to the case when the observations are i.i.d. and belong to a finite alphabet. Under these assumptions, Hoeffding showed that the test of (3.6) is in fact *asymptotically optimal* in terms of achieving the best performance w.r.t. error exponents. This result was generalized by Natarajan [57] to the case

when underlying probability distributions are Markovian. To the best of our knowledge, no further generalizations exist beyond this. Our choice of the goodness-of-fit test is loosely based on the optimality results of Hoeffding [56] and Natarajan [57]. In this thesis, we will not attempt to generalize their results to our setup.

3.4.2 The Estimation Problem

A key issue remains unresolved – that of estimating the above divergence-rate of (3.1) given access to only the traffic data $d_1, d_2, \ldots d_n$ and a reference arrival process $\mathbf{A}^n$. The statistics of the process $\mathbf{A}^n$ may sometimes need to be estimated by observing the network over large periods of time. To the best of our knowledge, no prior work in steganalysis or forensics literature has studied this problem rigorously. A first attempt was reported by Ezzeddine and Moulin [49], where the KL divergence-rate between the covertext and stegotext processes is approximated by the KL divergence between their respective *first-order* (empirical) marginals. Specifically, the divergence-rate estimate in (3.6) is replaced by $D(p_A \| \hat{p}_D)$, where

$$\hat{p}_D(d) = \frac{1}{n} \#\{i : d_i = d\} \tag{3.7}$$

is the empirical pmf (histogram) of observed interdeparture times. This approach makes calculations simpler, but is not accurate when the observations are not i.i.d., typically the case in covert timing channels.[2] Further, when the underlying data are not i.i.d., estimating higher order distributions is a challenging task, due to the 'curse of dimensionality' problem that sets in when the alphabet size is large. In fact, as

[2]Consider, for example, two first-order Markov processes defined as follows: $X_n = X_{n-1} \oplus U_n$ initialized with $X_0 = 0$, and $Y_n = Y_{n-1} \oplus V_n$ initialized with $Y_0 = 0$, where U_n and V_n are i.i.d. Bernoulli random variables with parameters α and β respectively, independent of everything else, and $\oplus$ denotes modulo 2 addition. In this case, the KL divergence between the two processes' first-order marginals is zero, while the divergence-rate is $D(\text{Bernoulli}(\alpha) \| \text{Bernoulli}(\beta)) = \alpha \log \frac{\alpha}{\beta} + (1-\alpha) \log \frac{1-\alpha}{1-\beta}$, which is nonzero for any $\alpha \neq \beta$.

the alphabet size becomes infinite, the counting estimator of (3.7) produces arbitrarily large variance in the estimator of KL divergence. For the remainder of this chapter, we will therefore explore the following problem in greater depth:

> Given realizations from two processes $\mathbf{X}^n$ and $\mathbf{Y}^n$, with *unknown* underlying probability laws P and Q, i.e., given samples $\{a_i\}_{i=1}^n$ and $\{d_j\}_{i=1}^n$, how does one estimate the KL divergence-rate between the two?

In a timing channel context, $\mathbf{X}^n$ and $\mathbf{Y}^n$ refer to the covertext ($\mathbf{A}^n$) and stegotext ($\mathbf{D}^n$) processes respectively. We shall specifically be interested in techniques that apply to P and Q corresponding to queue-based stegocodes for timing channels, if not more general scenarios.

3.5 Related Work

The above problem of estimating the KL divergence-rate of sources whose distributions are not explicitly known had gained the attention of information theorists in the early 1990s. Much of the foundation was provided by earlier literature on entropy-rate estimators that began to appear starting in the late 1970s with the publication of the celebrated Lempel-Ziv (LZ) source coding algorithm [58]. However, despite KL divergence being a fundamental information measure, its estimation problem has received relatively little attention; we list some of the known work here in Sections 3.5.1 and 3.5.2. At this point, it may be helpful to review the notation and definitions from Section 3.1.

3.5.1 Finite Alphabets

Ziv and Merhav [59] applied the idea of LZ parsing to divergence estimation. They developed a scheme to estimate the divergence between two finite-alphabet, finite-order stationary Markov sources, and

established the consistency of the estimator under the assumption that the observations are generated by independent finite alphabet, Markov sources. Their estimator is *universal* in the sense of not depending on the order or any other information about the transition probability matrices of the sources. The basic idea behind this and other similar estimators for divergence-rates is easily understood from the point of view of entropy-rate estimation – wherein the fundamental underpinning is provided by the celebrated result in information theory that establishes entropy-rate of a source as the compression limit of any lossless coder. Hence, the entropy-rate of a source can be indirectly measured via a 'good' source coding algorithm, by studying its compression properties.

Among the early works, Benedetto *et al.* [60] proposed an estimator loosely based on LZ compression. Cai *et al.* then studied two divergence estimation algorithms [61], both of which are motivated by techniques in data compression. The first estimator uses the Burrow-Wheeler transform (BWT) [62], while the second estimator uses the context tree weighting (CTW) method [63]. Consistency properties of both estimators are established under the assumption that both sources are possibly dependent, stationary ergodic, finite alphabet Markov sources. Some experimental results have also been reported by Dawy *et al.* [64] using the CTW method for classification of binary sequences.

All of the above is however only for finite alphabet sources, and does not directly carry over to our timing channels setup wherein the underlying processes are over an infinite (but countable) alphabet. As discussed in Section 3.4, this is a nontrivial extension. Further, by virtue of the queue action, the stegotext process (interdeparture times) is not necessarily Markov (of any order), and can potentially possess long range dependencies (depending on other factors such as the arrival rate and the service rate of the queue). In this context, we ideally need divergence estimators that work for countable alphabet processes, with

possibly long-range dependencies. To the best of our knowledge, no previous research has explored these scenarios. The 'closest' variants to our problem can be found in [65–70], and are discussed next.

3.5.2 Countable Alphabets

As is the case with several divergence-rate estimators proposed in the literature, much of the early groundwork for the countable alphabet case was made in the area of entropy-rate estimation. Let us begin with a review of some known results concerning a stationary process $\mathbf{X}$, defined on a countable alphabet.

For $w \geq 1$, let $L_w = L_w(\mathbf{X})$ denote the minimum length k of the string X_0^{k-1} that starts at time 0 and does not appear as a contiguous substring within the window of past w samples X_{-w}^{-1}. Alternatively, we may define L_w as follows:

$$L_w \triangleq 1 + \max\{l : 0 \leq l \leq n, X_0^{l-1} = X_{-j}^{-j+l-1} \text{ for some } l \leq j \leq w\} \quad (3.8)$$

Wyner and Ziv [71] showed that for every ergodic process, the quantity $L_w / \log w$ converges to $1/H'(P)$ in probability. This above result was strengthened by Ornstein and Weiss [72] to establish almost sure convergence, i.e.,

$$\frac{L_w}{\log w} \to \frac{1}{H'(P)} \quad P \text{ a.s.} \quad (3.9)$$

At about the same time, Grassberger [65] suggested the following result, based on *averages* of match lengths, for i.i.d. processes with countable alphabet:

$$\frac{w \log w}{\sum_{i=1}^{w} L_w^{(i)}} \to H'(P) \quad P \text{ a.s.}, \quad (3.10)$$

where $L_w^{(i)} = L_w(T^i(\mathbf{X}))$, $T(\mathbf{X}) = \{X_{i+1}\}_{i \in \mathbb{Z}}$, the original sequence shifted by one time unit. Shields [66] showed that the above convergence does not extend to general ergodic processes, although it does for Markov

chains. Kontoyiannis and Suhov [67] extended this to a wider class of stationary processes with long range memory, and not necessarily Markov, and Quas [70] extended it further to certain processes with infinite alphabets and to random fields. Extensions of the above results to divergence-rates are almost nonexistent excepting in Kontoyiannis' thesis [69] where the following result is established for two independent processes **X** and **Y**:

Theorem 1 [69, Corollary 4.11]: Let **X** be a finite-valued stationary ergodic process with distribution P, **Y** be a stationary ergodic Markov chain with distribution Q, and assume that P is absolutely continuous w.r.t. Q (if not, the divergence-rate between the two processes is undefined). Then, we have

$$\frac{\tilde{L}_w}{\log w} \to \frac{1}{H'(P) + D'(P\|Q)} \quad P \times Q \text{ a.s.}, \tag{3.11}$$

where $H'(P)$ is the entropy-rate of **X**, $D'(P\|Q)$ is the divergence-rate between **X** and **Y**, and $\tilde{L}_w$ is a quantity defined similarly to L_w as follows:

$$\tilde{L}_w \triangleq \max\{l \geq 1 : Y_j^{j+l-1} = X_0^{l-1} \text{ for some } 1 \leq j \leq w\}. \tag{3.12}$$

In other words, $\tilde{L}_w$ is the longest string X_0^{l-1} with a match in Y_0^{W-1}.

For finite $\sigma^2 \triangleq \lim_{n\to\infty} \mathrm{Var}_P(-\log Q(X_1^n))$, [69] also showed that:

$$\frac{\tilde{L}_w - \frac{\log w}{H'(P)+D'(P\|Q)}}{\sqrt{\log w}} \xrightarrow{\mathscr{D}} \mathcal{N}\left(0, \frac{\sigma^2}{(H'(P)+D'(P\|Q))^3}\right) \tag{3.13}$$

$$\limsup_{w\to\infty} \frac{\tilde{L}_w - \frac{\log w}{H'(P)+D'(P\|Q)}}{\sqrt{2\log w \ln\ln\log w}} \overset{a.s.}{=} \frac{\sigma}{(H'(P)+D'(P\|Q))^{3/2}}, \tag{3.14}$$

thus establishing the pointwise rate of convergence for (3.11).

Now, under the assumptions of Theorem 1, (3.10) and (3.11) can be combined to show the following:

$$\frac{w \log w}{\sum_{i=1}^{w} \tilde{L}_w^{(i)}} - \frac{w \log w}{\sum_{i=1}^{w} L_w^{(i)}} \to D'(P\|Q) \quad P \times Q \text{ a.s.}, \tag{3.15}$$

where $\tilde{L}_w^{(i)} = \tilde{L}_w(T^i(\mathbf{X}))$. The proof follows from (3.11) and (3.9), that respectively establish the convergence of $\frac{\tilde{L}_w^{(i)}}{w \log w}$ and $\frac{L_w^{(i)}}{w \log w}$, $\forall i$, almost surely to $\frac{1}{H'(P)+D'(P\|Q)}$ and $\frac{1}{H'(P)}$ respectively. Hence, we have

$$\frac{\sum_{i=1}^{w} \tilde{L}_w^{(i)}}{w \log w} \to \frac{1}{H'(P)+D'(P\|Q)} \quad P \times Q \text{ a.s.} \tag{3.16}$$

$$\frac{\sum_{i=1}^{w} L_w^{(i)}}{w \log w} \to \frac{1}{H'(P)} \quad P \text{ a.s.} \tag{3.17}$$

and the result in (3.15) follows.

While close enough, (3.15) does not directly carry over to $\mathbf{X}$ and $\mathbf{Y}$ that correspond to the covertext and stegotext processes of timing channels.

One reason is that (3.11) was proved only for finite alphabets, and Markovian $\mathbf{Y}$. However, a careful examination of the arguments presented in [69] reveals that the result in (3.11) (and hence (3.15)) remains true for countably infinite valued processes $\mathbf{X}$ and $\mathbf{Y}$, and for general (stationary, ergodic) k^{th} order Markov processes $\mathbf{Y}$, under the additional assumptions that the entropy-rate of $\mathbf{X}$ and the divergence-rate between $\mathbf{X}$ and $\mathbf{Y}$ are both finite. To be more specific, the extension to countable alphabets works because the proof of (3.11) mainly relies on two results:

(A1) the Shannon-McMillan-Breiman theorem [73–75]:

$$-\frac{1}{n} \log P(X_1^n) \to H'(P) \quad \text{in probability}, \tag{3.18}$$

(A2) a strong approximation result shown in [69, Theorem 4.1] which

states the following:

$$\lim_{n\to\infty} \frac{1}{\sqrt{n}} \Big(\log W_n - \log[1/Q(X_1^n)] \Big) = 0 \quad P \times Q \text{ a.s.} \tag{3.19}$$

where W_n is the waiting time until X_1^n appears in Y_1^∞. Equivalently,

$$W_n \triangleq \inf\{k \geq 1 : Y_k^{k+n-1} = X_1^n\}. \tag{3.20}$$

It is well known that (A1) is true for countable alphabets if $H'(P) < \infty$; this was first shown in [76]. (A2) also generalizes to countable alphabets if $D'(P\|Q) < \infty$, and this can be inferred by inspection of [69, Theorem 4.1]. There is however a second and possibly much trickier issue – the interdeparture process $\mathbf{D}^n$ is not necessarily Markovian of any order. To clear this hurdle, we ask if it is possible to extend the estimator of (3.15) to timing channels?

3.6 Conjecture

We conjecture that (3.15) continues to hold even when $\mathbf{Y}$ is not finite order Markov, but instead the process observed at the output of a queue-based stegocoder. We are currently working on a formal proof to establish this result.

In the following section, we study the convergence properties of the proposed estimator (3.15) empirically on three different queue-based timing channel codes.

3.7 Proposed Estimator

Assuming truth of the above conjecture, given $\{a_i\}_{i=-n}^n$ and $\{d_i\}_{i=-n}^n$ (finite length realizations of the covertext and stegotext processes respectively), we propose the following sequence of divergence-rate esti-

mators:

$$\hat{D}'_w \triangleq \frac{w \log w}{\sum_{i=1}^{w} \tilde{M}_w^{(i)}} - \frac{w \log w}{\sum_{i=1}^{w} M_w^{(i)}}, \quad w \geq 0 \tag{3.21}$$

where $\tilde{M}_w^{(i)}, M_w^{(i)}$ are defined analogously to $\tilde{L}_w^{(i)}, L_w^{(i)}$, with appropriate modifications to curtail searches within the finite boundaries of $\{a_i\}_{i=-n}^{n}$ and $\{d_i\}_{i=-n}^{n}$. If $\tilde{M}_w^{(i)}, M_w^{(i)}$ do not admit numerical values (e.g., no match exists), we set it equal to the length of the search window w.

3.8 Experimental Results

In this section, we evaluate the performance of the estimator proposed in (3.21) on three simple stegocodes for covert timing channels. For all scenarios, we model the arrival process $\mathbf{A}^n$ to be i.i.d. with a Geo(λ) distribution:

$$p_A(a) = (1-\lambda)^{a-1}\lambda, \quad a = 1, 2, 3, \ldots \tag{3.22}$$

The stegocoding operations are described below.

3.8.1 Cybenko's Non-Queue-Based Code

This code modifies an interarrival packet time A_i into an interdeparture time to embed a binary bit $b_i \in \{0, 1\}$ according to the following formula [38]:

$$D_i = \begin{cases} 2\lfloor A_i/2 \rfloor + S'_i, & M_i = 0 \\ 2\lfloor A_i/2 \rfloor + S'_i + 1, & M_i = 1, \end{cases} \tag{3.23}$$

where S'_i is an odd random number, whose pmf is given in (3.5). Evidently, the expected value of D_i is strictly greater than the expected value of A_i, and this creates a lag with linearly increasing mean as the transmission time increases (Figure 3.3). This is not desirable for two reasons: one, it constrains the stegocoder to actively transmit only over

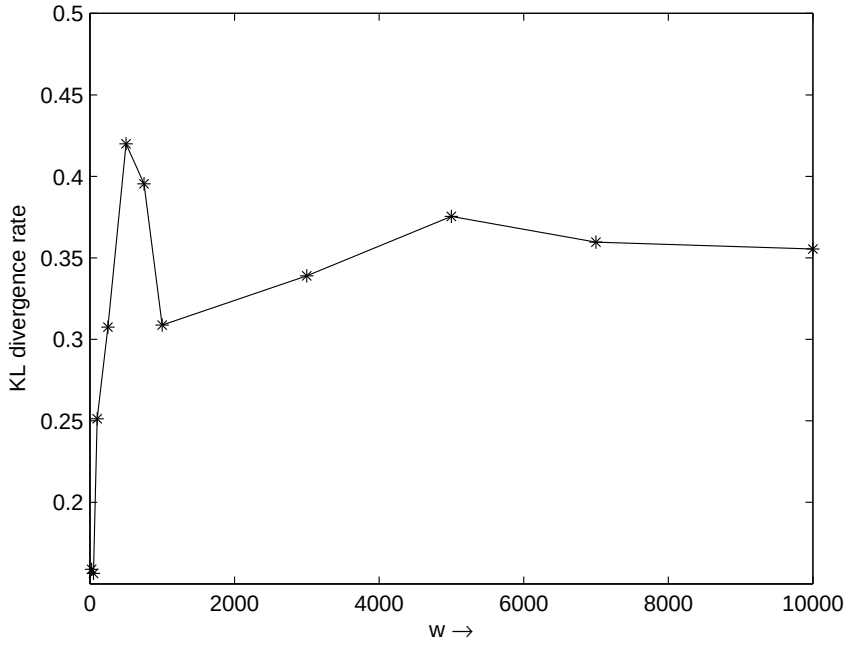


Figure 3.4: KL divergence-rate estimate as a function of window length w for Cybenko's code (Section 3.8.1).

brief periods of time (to keep the overall delay small), and hence results in an inefficient exploitation of the covert channel; and two, it makes the covert transmission easily detectable.

To quantify the detectability of the code, we use the estimator of (3.15) to estimate the KL divergence-rate between the processes $\mathbf{A}^n$ and $\mathbf{D}^n$. Figure 3.4 plots the estimates for various values of the window parameter w. At $w = 10^4$, the estimated value of the divergence-rate is ≈ 0.36. Further, we provide the histograms corresponding to the estimates for various values of w in Figure 3.5. As suggested by the histograms, for $w \to \infty$, the estimates seem to converge in distribution to a Gaussian. It, however, remains to be seen whether the estimates indeed converge to the true value of the divergence-rate, and if it does so in any stronger sense of convergence. Another point to note is that the estimates are far off (from the eventual values) when w is small. This can be attributed to the high variances in match lengths one will see when the

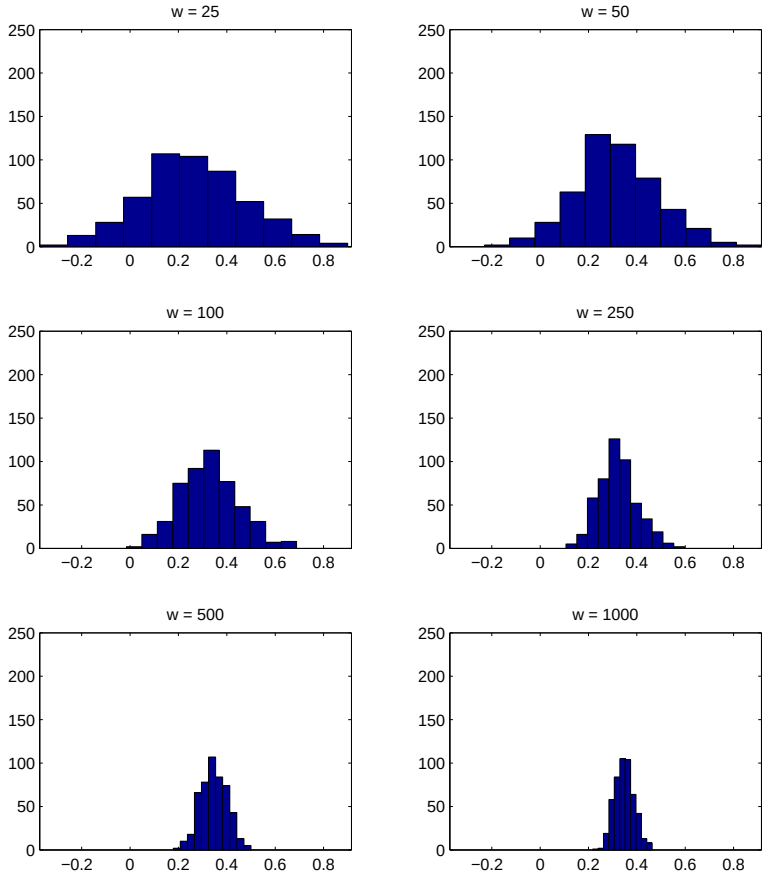


Figure 3.5: Histograms for the divergence-rate estimates of (3.15), indexed by w, for Cybenko's code (Section 3.8.1).

search window size is small.

3.8.2 A Simple Queue-Based Stegocode

As explained earlier, the queue-based stegocode of (3.4) does a good job at circumventing the drift problem but is not perfectly secure. A quick calculation would reveal that the interdeparture times, unlike the interarrival times, are *not* geometrically distributed, and thus, the divergence-rate between A^n and D^n cannot be made arbitrarily small. Our estimates of the divergence-rate for various values of w is given

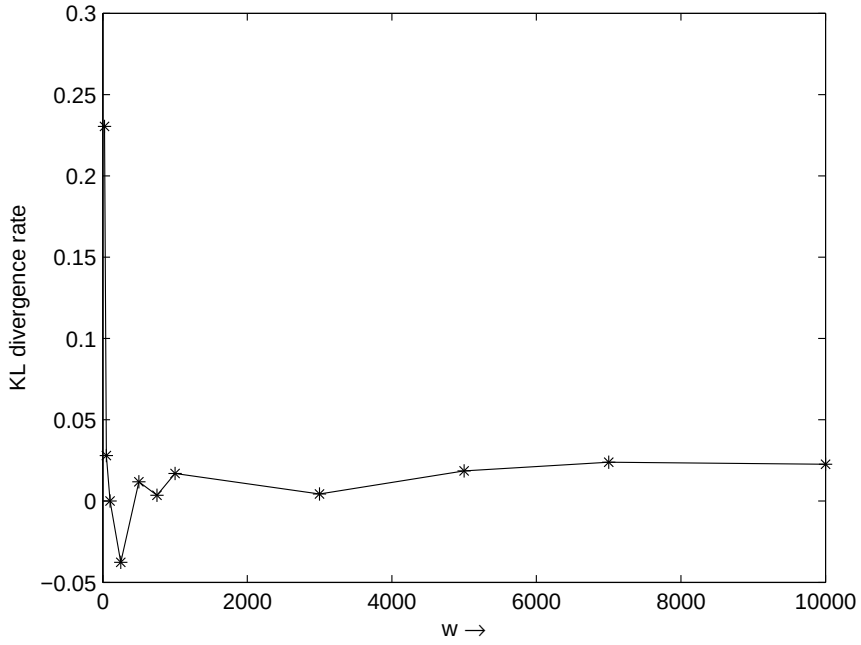


Figure 3.6: KL divergence-rate estimate as a function of window length w for simple queue-based code (Section 3.8.2).

in Figure 3.6. From the graph, we infer that this code outperforms Cybenko's, and the divergence-rate estimate corresponding to $w = 10^4$ is ≈ 0.03.

3.8.3 Stochastic Queue-Based Stegocode

Stochastic queue-based stegocodes trade off detectability for decoding performance. A simple stochastic code was proposed by Moulin [48] that embeds bits via a similar queue action as before, but introduces a twist via randomization for matching the distribution of the interdeparture times to a geometric distribution. The randomization is done as follows:

$$D_i = \begin{cases} W_i + S_i' & \text{w.p. } \frac{\mu}{2-\mu} \qquad W_i + M_i \text{ even} \\ W_i + S_i' + 1 & \text{w.p. } 1 - \frac{\mu}{2-\mu} \qquad W_i + M_i \text{ even} \\ W_i + S_i' & \text{w.p. } 1 \qquad W_i + M_i \text{ odd,} \end{cases} \tag{3.24}$$

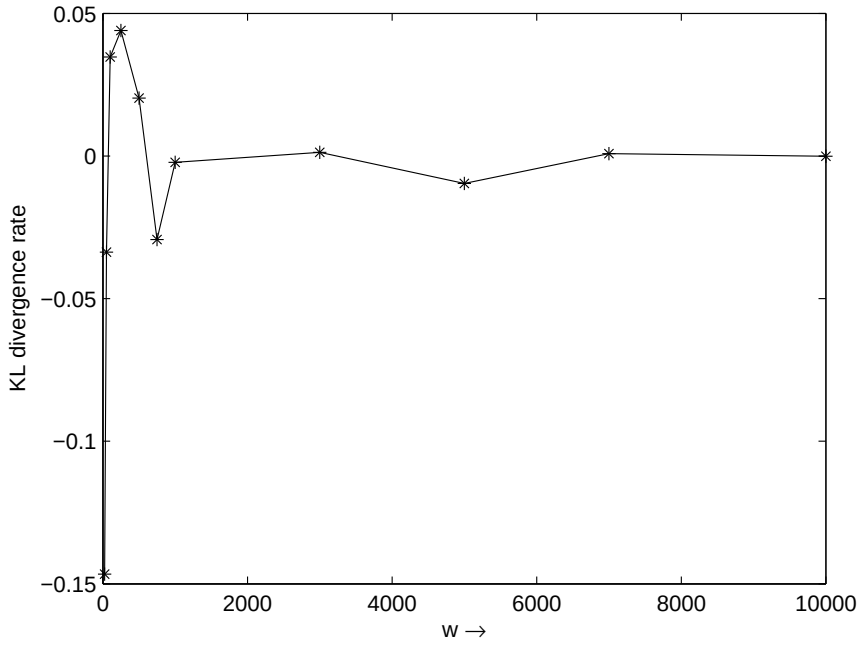


Figure 3.7: KL divergence-rate estimate as a function of window length w for stochastic queue-based code (Section 3.8.3).

where S'_i is an odd random number drawn from the distribution given in (3.5). The above randomization reduces the information hiding rate to less than one bit per transmission, but perfectly matches the marginals of the covertext and stegotext processes. This seems like a promising improvement from a security standpoint. It is not however immediately obvious if this also results in an arbitrarily small value for the divergence-rate. From Figure 3.7, the value of the divergence-rate, corresponding to $w = 10^4$, is $\approx 10^{-3}$ – clearly the least among the three codes that we have seen thus far.

Chapter 4

Future Directions

The results presented in this work are encouraging and suggest several additional problems to explore. Of these, perhaps the most compelling (from a commercial standpoint) would be to investigate the resilience of this family of decoders to geometric distortions (also referred in the literature as desynchronizations) such as zooming, spatial warping and rotation. In principle, it is straightforward to construct a (potentially naïve) decoder that attempts to perform probabilistic inference under geometric transformations of the host. However, it is not clear what would be the best approach to do so – specifically, the way to go about handling "messy" loops and factors is not immediately clear. Any significant breakthrough on that front is bound to project iterative graph-based decoding strategies introduced in this work as a generic and powerful tool to cope with a remarkably wide range of commonly encountered signal distortions.

Another avenue for exploration would be to tap the power of message-passing based decoders when the distortion model is unknown. If we know that the distortion channel has introduced one among a few K possible transformations, we can perform an exhaustive search within the confines of K possible parameterizations of the channel. However, the problem is compounded significantly if K is large, or potentially even infinite. In such cases, a better approach might be to perform some preprocessing that aims to learn the structure of the underlying graph based on samples of the received signal. We have some preliminary ideas in this direction.

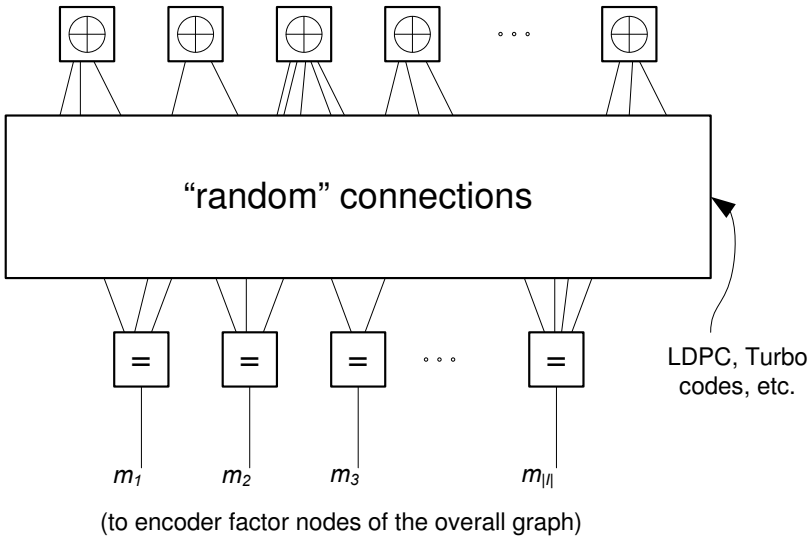


Figure 4.1: More powerful alternatives to repetition codes (such as LDPC codes) allow for lower probabilities of decoding error. Due to space constraints, we omit showing the remainder of the graph with the host, distortion model, etc. Note that the "$\oplus$" blocks above are similar to the zero-sum constraint nodes introduced in Section 2.4, the only difference being that the addition is done modulo two. Message updates for this section of the graph can typically be processed in batches, i.e., all messages corresponding to the upper set of nodes in one step, followed by the lower ones.

Finally, while we use simple repetition codes for channel coding in our setup, the decoder model proposed in Section 2.4 offers the potential to construct efficient blind watermark decoders that incorporate, in the Bayesian inference setup, powerful error-correcting codes (e.g., turbo or LDPC codes), whose decoding is also done via belief propagation on an appropriate factor graph (see Figure 4.1) [45, 77]. We believe such a decoder, that would fully combine the power of graph-based distortion parameters' estimation (for a wide range of distortions), and a powerful, interleaved error-correcting code for achieving decoding error probabilities much lower than what repetition codes are able to offer, could turn out be an important milestone for blind watermark decoding.

As for the work on divergence-rate estimators, it would be helpful to formally prove the conjecture in Section 3.6. Such a result would establish, for the first time, existence of *provably consistent*, universal estimators for the KL divergence-rate between covertext and stegotext processes in queue-based timing channels. It would also be interesting to explore if the discussion on optimality in Section 3.4.1 can be formalized. As far as we know, this has not been done before for cases other than i.i.d. and Markov sources.

Another equally important line of research could explore construction of estimators with convergence properties faster than that of the match-length based estimator introduced in this thesis. This might lead to interetsing tradeoffs between convergence speed, bias and variance of the estimators.

As a concluding note, this thesis has inspired us to believe that the roles of probabilistic inference and statistical estimation theory are heavily under-tapped in the data-hiding community, and that these could very well hold crucial keys to bridging many an existing gap between theory and practice in several data-hiding problems. We look forward to seeing interesting developments in this direction in the years to come.

Appendix A

Estimating MRF Parameters

Here, we provide details about the estimation of Markov random field (MRF) parameters in (2.12). For convenience, we will first rewrite (2.12) as follows:

$$p(\mathbf{s}) = \frac{1}{Z_1} \exp\{-U(\mathbf{s})\}, \tag{A.1}$$

where

$$U(\mathbf{s}) = \sum_{i \in \mathscr{I}} \frac{(s_i - \mu_s)^2}{2\sigma_{sa}^2} + \frac{(s_i - s_{iE})^2}{8\sigma_{sb}^2} + \frac{(s_i - s_{iN})^2}{8\sigma_{sb}^2}. \tag{A.2}$$

An obvious choice for estimating the parameters μ_s, σ_{sa} and σ_{sb} would be the maximum-likelihood (ML) method, wherein we may maximize the log likelihood function

$$\begin{aligned} l(\mu_s, \sigma_{sa}, \sigma_{sb}) &= \ln p(\mathbf{s}) && \text{(A.3)} \\ &= -\ln Z_1 - U(\mathbf{s}) && \text{(A.4)} \end{aligned}$$

over μ_s, σ_{sa} and σ_{sb}. However, computation of the partition function

$$Z_1 = \int_{\mathbb{R}^{|\mathscr{I}|}} \exp\{-U(\mathbf{s})\}\, d\mathbf{s} \tag{A.5}$$

involves a computationally prohibitive integral over all possible configurations $\mathbf{s} \in \mathbb{R}^{|\mathscr{I}|}$, which is typically impossible to calculate for all practical purposes as $|\mathscr{I}|$ is usually large. The ML method can therefore not be implemented, and we resort to a pseudo-maximum likelihood approach that exploits the local characteristics of the MRF and yields good approximations to the estimates we seek (also known in the literature as 'coding method' [78]).

A coding is a set of sites which are conditionally independent given their own neighborhood. For instance, consider the MRF with first-order neighborhood of (A.2). By subsampling $\mathscr{I}$ according to a quincunx scheme, we obtain two codings $\mathscr{I}_\circ$ and $\mathscr{I}_\bullet$ whose elements are respectively circles and bullets in the figure below.

$$\begin{array}{ccccc} \circ & \bullet & \circ & \bullet & \dots \\ \bullet & \circ & \bullet & \circ & \dots \\ \circ & \bullet & \circ & \bullet & \dots \\ \bullet & \circ & \bullet & \circ & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

Based on the above codings, we can also now define configurations $\mathbf{s}_\circ \triangleq \{s_i : i \in \mathscr{I}_\circ\}$ and $\mathbf{s}_\bullet \triangleq \{s_i : i \in \mathscr{I}_\bullet\}$. It is now easy to note that conditioned on $\mathbf{s}_\circ$, the joint probability distribution of $\mathbf{s}_\bullet$ takes the following elegant product form:

$$p(\mathbf{s}_\bullet|\mathbf{s}_\circ) = \prod_{i\in\mathscr{I}_\bullet} p(s_i|\mathbf{s}_{\nu(i)}) \tag{A.6}$$

$$= \prod_{i\in\mathscr{I}_\bullet} \psi\left(\frac{\frac{\mu_s}{\sigma_{sa}^2} + \frac{s_{iE}+s_{iW}+s_{iN}+s_{iS}}{4\sigma_{sb}^2}}{\frac{1}{\sigma_{sa}^2}+\frac{1}{\sigma_{sb}^2}}, \frac{1}{\frac{1}{\sigma_{sa}^2}+\frac{1}{\sigma_{sb}^2}}\right), \tag{A.7}$$

where $\nu(i) = \{iE, iW, iN, iS\}$ denotes the locations immediately to the east, west, north and south of i (or in short, the neighborhood of i). We now define a local log-likelihood function

$$l(\mu_s, \sigma_{sa}, \sigma_{sb}|s_i, \mathbf{s}_{\nu(i)}) = \ln p(s_i|\mathbf{s}_{\nu(i)}), \tag{A.8}$$

and obtain a computationally tractable optimization problem:

$$(\hat{\mu}_{s\bullet}, \hat{\sigma}_{sa\bullet}, \hat{\sigma}_{sb\bullet}) = \underset{\mu_s,\sigma_{sa},\sigma_{sb}}{\operatorname{argmax}} \sum_{i\in\mathscr{I}_\bullet} l(\mu_s, \sigma_{sa}, \sigma_{sb}|s_i, \mathbf{s}_{\nu(i)}), \tag{A.9}$$

for which numerical solutions can be found. Further, we can also obtain another set of estimates $(\hat{\mu}_{s\circ}, \hat{\sigma}_{sa\circ}, \hat{\sigma}_{sb\circ})$ by switching the roles of $\mathbf{s}_\bullet$ and $\mathbf{s}_\circ$. The final pseudo ML estimates of the MRF parameters are obtained as an average of the above two estimates. Note that the estimates $(\hat{\mu}_{s\bullet}, \hat{\sigma}_{sa\bullet}, \hat{\sigma}_{sb\bullet})$ and $(\hat{\mu}_{s\circ}, \hat{\sigma}_{sa\circ}, \hat{\sigma}_{sb\circ})$ are highly correlated as the configurations $\mathbf{s}_\bullet$ and $\mathbf{s}_\circ$ are dependent. This procedure is quite popular due to its implementation efficiency, and also due to its theoretical advantages, including convergence in probability to the true parameter values as $|\mathscr{I}| \to \infty$ [78].

If instead we had an MRF with second-order neighborhoods, a similar procedure would be used, but with four codings $\mathscr{I}_\bullet, \mathscr{I}_\circ, \mathscr{I}_\bigstar$ and $\mathscr{I}_\triangle$ defined below. Here, the samples of $\mathbf{s}_\bullet$ defined over $\mathscr{I}_\bullet$ are conditionally independent given the other three codings, etc.

$$
\begin{array}{ccccccc}
\bullet & \triangle & \bullet & \triangle & \bullet & \triangle & \dots \\
\bigstar & \circ & \bigstar & \circ & \bigstar & \circ & \dots \\
\bullet & \triangle & \bullet & \triangle & \bullet & \triangle & \dots \\
\bigstar & \circ & \bigstar & \circ & \bigstar & \circ & \dots \\
\bullet & \triangle & \bullet & \triangle & \bullet & \triangle & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}
$$

References

[1] M. D. Swanson, M. Koboyashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064–1087, June 1998.

[2] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding – a survey," *Proceedings of the IEEE*, pp. 1062–1078, July 1999.

[3] P. Moulin and R. Koetter, "Data-hiding codes," *Proceedings of the IEEE*, vol. 93, no. 12, pp. 2083–2126, Dec. 2005.

[4] M. Barni and F. Bartolini, *Watermarking Systems Engineering*. New York, NY: Marcel Dekker, 2004.

[5] I. J. Cox, M. L. Miller, and J. A. Miller, *Digital Watermarking*. San Francisco, CA: Morgan-Kaufmann, 2002.

[6] J. Eggers and B. Girod, *Informed Watermarking*. Boston, MA: Kluwer, 2002.

[7] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding, Steganography and Watermarking – Attacks and Countermeasures*. Boston, MA: Kluwer, 2001.

[8] S. Katzenbeisser and F. A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*. Norwood, MA: Artech House, 2000.

[9] P. Moulin and J. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 563–593, 2003.

[10] F. Balado, K. M. Whelan, G. C. M. Silvestre, and N. J. Hurley, "Joint iterative decoding and estimation for side-informed data hiding," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 4006–4019, Oct. 2005.

[11] R. L. Lagendijk and I. D. Shterev, "Estimation of attacker's scale and noise variance for QIM-DC watermark embedding," in *IEEE International Conference on Image Processing*, vol. 1, Oct. 2004, pp. 24–27.

[12] I. D. Shterev and R. L. Lagendijk, "Amplitude scale estimation for quantization-based watermarking," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4146–4155, Nov. 2006.

[13] F. Perez-Gonzalez, C. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation: a high-rate data-hiding method invariant to gain attacks," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3960–3975, Oct. 2005.

[14] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[15] M. Kutter, "Watermarking resisting to translation, rotation and scaling," in *Proc. SPIE, Boston*, vol. 3528, Jan. 1999, pp. 423–431.

[16] J. J. ÓRuanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 303–317, May 1998.

[17] F. Perez-Gonzalez and C. Mosquera, "Quantization-based data hiding robust to linear-time-invariant filtering," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 137–152, June 2008.

[18] M. L. Miller, G. J. Doerr, and I. J. Cox, "Dirty-paper trellis codes for watermarking," in *IEEE International Conference on Image Processing*, vol. 2, no. 2, 2002, pp. 129–132.

[19] M. L. Miller, G. J. Doerr, and I. J. Cox, "Applying informed coding and embedding to design a robust high-capacity watermark," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 792–807, June 2004.

[20] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1123–1129, June 2000.

[21] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for

images," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 767–782, May 2001.

[22] M. Álvarez Rodríguez and F. Pérez-González, "Analysis of pilot-based synchronization algorithms for watermarking of still images," *Signal Processing: Image Communication*, vol. 17, pp. 611–633, Sep. 2002.

[23] P. Moulin and A. Ivanović, "The Fisher information game for optimal design of synchronization patterns in blind watermarking," in *IEEE International Conference on Image Processing*, vol. 2, Oct. 2001, pp. 550–553.

[24] P. Moulin, "Embedded-signal design for channel parameter estimation. Part I: Linear embedding," in *IEEE Statistical Signal Processing Workshop*, Sep. 2003, pp. 38–41.

[25] P. Moulin, "Embedded-signal design for channel parameter estimation. Part II: Quantization embedding," in *IEEE Statistical Signal Processing Workshop*, Sep. 2003, pp. 42–45.

[26] J. J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod, "Scalar Costa scheme for information embedding," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1003–1019, Apr. 2003.

[27] I. D. Shterev and R. L. Lagendijk, "Maximum likelihood amplitude scale estimation for quantization-based watermarking in the presence of dither," in *SPIE Security, Steganography, Watermarking Multimedia Contents VII*, Jan. 2005, pp. 516–527.

[28] P. Moulin, A. Briassouli, and H. Malvar, "Detection-theoretic analysis of desynchronization attacks in watermarking," in *Proc. 14th International Conference on Digital Signal Processing*, July 2002, pp. 77–84.

[29] K. M. Whelan, F. Balado, G. C. M. Silvestre, and N. J. Hurley, "PLL-based synchronization of dither modulation data hiding," in *IEEE International Conference on Accoustics, Speech and Signal Processing*, May 2006, p. II.

[30] M. Feder and A. Lapidoth, "Universal decoding for channels with memory," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1726–1745, Sep. 1998.

[31] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2148–2177, Oct. 1998.

[32] P. Moulin, "On the optimal structure of watermark decoders under desynchronization attacks," in *IEEE International Conference on Image Processing*, Oct. 2006, pp. 2589–2592.

[33] P. Moulin, "Universal decoding of watermarks under geometric attacks," in *IEEE International Symposium on Information Theory*, July 2006, pp. 2353–2357.

[34] S. Sadasivam and P. Moulin, "On estimation accuracy of desynchronization attack channel parameters," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 284–292, Sep. 2009.

[35] B. Lampson, "A note on the confinement problem," in *Comm. ACM*, vol. 16, Oct. 1973, pp. 613–615.

[36] J. Giles and B. Hajek, "An information-theoretic and game-theoretic study of timing channels," *IEEE Trans. on Info. Theory*, vol. 48, no. 9, pp. 2455–2477, Sep. 2002.

[37] S. Cabuk, C. E. Brodley, and C. Shields, "IP covert timing channels: Design and detection," in *11^{th} ACM Conf. on Computer and Communications Security (CCS)*, New York, NY, 2004, pp. 178–187.

[38] V. Berk, A. Giani, and G. Cybenko, "Detection of covert channel encoding in network packet delays," Dept. of Comp. Sci., Dartmouth College, Tech. Rep. TR2005536, 2005.

[39] S. L. Lauritzen, *Graphical Models*. Oxford, UK: Oxford University Press, 1996.

[40] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann, 1988.

[41] B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*. Cambridge, MA: MIT Press, 1998.

[42] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory, Special Issue on Codes on Graphs and Iterative Algorithms*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[43] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Advances in Neural Information Processing Systems*, vol. 13, Dec. 2000, pp. 689–695.

[44] M. I. Jordan, "Graphical models," *Statistical Science, Special issue on Bayesian statistics*, vol. 19, pp. 140–155, 2004.

[45] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[46] J. H. G. Dauwels, "On graphical models for communications and machine learning: Algorithms, bounds, and analog implementation," Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich, May 2006.

[47] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[48] P. Moulin, "Queue-based codes for timing channels," in *Info. Theory and Appl. (ITA) Workshop*, San Diego, CA, Jan. 2008.

[49] I. Ezzeddine and P. Moulin, "Achievable rates for queue-based timing stegocodes," in *Proc. IEEE Info. Theory Workshop*, Sicily, Italy, Oct. 2009, pp. 379–383.

[50] D. Lindley, "The theory of queues with a single server," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, no. 2, pp. 277–289, 1952.

[51] G. Shah, A. Molina, and M. Blaze, "Keyboards and covert channels," in *Proc. USENIX Security*, 2006, pp. 59–75.

[52] X. Luo, E. Chan, and R. Chang, "Tcp covert timing channels: Design and detection," in *IEEE International Conference on Dependable Systems and Networks with FTCS and DCC*, 2008, pp. 420 –429.

[53] K.-S. Song, "Goodness-of-fit tests based on Kullback-Leibler discrimination information," *IEEE Trans. on Info. Theory*, vol. 48, no. 5, pp. 1103–1117, 2002.

[54] B. Senoglu and B. Surucu, "Goodness-of-fit tests based on Kullback-Leibler information," *IEEE Trans. on Reliability*, vol. 53, no. 3, pp. 357–361, 2004.

[55] O. Dabeer, K. Sullivan, U. Madhow, S. Chandrasekharan, and B. S. Manjunath, "Detection of hiding in the least significant bit," *IEEE Trans. on Signal Proc., 1st supplement on Secure Media*, vol. 52, no. 10, pp. 3046–3058, Oct. 2004.

[56] W. Hoeffding, "Asymptotically optimal tests for multinominal distributions," *Ann. Math. Stat.*, vol. 36, pp. 369–408, 1965.

[57] S. Natarajan, "Large deviations, hypotheses testing, and source coding for finite Markov chains," *IEEE Transactions on Information Theory*, vol. 31, no. 3, pp. 360–365, May 1985.

[58] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Trans. on Info. Theory*, vol. 24, no. 5, pp. 530–536, Sep. 1978.

[59] J. Ziv and N. Merhav, "A measure of relative entropy between individual sequences with application to universal classification," *IEEE Trans. on Info. Theory*, vol. 39, no. 4, pp. 1270–1279, July 1993.

[60] D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping," *Physical Review Letters*, vol. 88, no. 4, Jan. 2002.

[61] H. Cai, S. R. Kulkarni, and S. Verdu, "Universal divergence estimation for finite alphabet sources," *IEEE Trans. on Info. Theory*, vol. 52, no. 8, pp. 3456–3475, Aug. 2006.

[62] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," Digital Systems Research Center, Tech. Rep. 124, 1994.

[63] F. M. J. Williems, Y. M. Shtarkov, and T. J. Tjalkens, "The context tree weighting method: Basic properties," *IEEE Trans. on Info. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[64] Z. Dawy, J. Hagenauer, and A. Hoffmann, "Implementing the context tree weighting method for content recognition," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2004, p. 536.

[65] P. Grassberger, "Estimating the information content of symbol sequences and efficient codes," *IEEE Trans. on Info. Theory*, vol. 35, pp. 669–675, May 1989.

[66] P. C. Shields, "Entropy and prefixes," *Ann. Prob.*, vol. 20, pp. 403–409, 1992.

[67] I. Kontoyiannis and Y. M. Suhov, *Probability Statistics and Optimization: A tribute to Peter Whittle (F. P. Kelly ed.)*. Chichester, England: Wiley, 1994, ch. Prefixes and the entropy rate for long-range sources, pp. 89–98.

[68] I. Kontoyiannis, P. H. Algoet, Y. M. Suhov, and A. J. Wyner, "Nonparametric entropy estimation for stationary processes and random fields, with applications to English text," *IEEE Trans. Info. Theory*, vol. 44, no. 3, pp. 1319–1327, May 1998.

[69] I. Kontoyiannis, "Recurrence and waiting times in stationary processes, and their applications in data compression," Ph.D. dissertation, Stanford University, Palo Alto, CA, May 1998.

[70] A. N. Quas, "An entropy estimator for a class of infinite alphabet processes," *Theory Prob. Appl.*, vol. 43, no. 3, pp. 496–507, 1998.

[71] A. D. Wyner and J. Ziv, "Some asymptotic properties of a stationary ergodic data source with applications to data compression," *IEEE Trans. on Info. Theory*, vol. 35, pp. 1250–1258, Nov. 1989.

[72] D. S. Ornstein and B. Weiss, "Entropy and data compression schemes," *IEEE Transactions on Information Theory*, vol. 39, pp. 78–83, Jan. 1993.

[73] C. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423,623–656, 1948.

[74] B. McMillan, "The basic theorems of information theory," *Ann. Math. Stat.*, vol. 24, pp. 196–219, 1953.

[75] L. Breiman, "The individual ergodic theorem of information theory," *Ann. Math. Stat.*, vol. 28, pp. 809–811, 1957.

[76] K. L. Chung, "A note on the ergodic theorem of information theory," *Ann. Math. Stat.*, vol. 32, pp. 612–614, 1961.

[77] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge, UK: Cambridge University Press, 2008.

[78] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society B*, vol. 36, no. 2, pp. 192–236, 1974.

Author's Biography

Shankar Sadasivam received his B.Tech. in Electrical Engineering from Indian Institute of Technology, Madras, in 2005, and M.S. in Electrical and Computer Engineering (ECE) from University of Illinois at Urbana-Champaign, in 2008. He is currently pursuing the Ph.D. degree in the ECE department at Illinois. His research interests include signal processing, communications, estimation theory, and machine learning. During the Fall 2007 and Spring 2008 semesters, he served as a teaching assistant for graduate level courses on information theory, and signal detection and estimation theory, respectively. He spent the summer of 2007 at IBM Research, NY, working on problems in the areas of optimization and workforce scheduling, and the summer of 2004 at École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, working on the design of pipelined analog-to-digital converters.