INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700  800/521-0600
RICE UNIVERSITY

TRANSFORM-DOMAIN MODELING OF NONGAUSSIAN AND $1/f$ PROCESSES

by

Matthew S. Crouse

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

DOCTOR OF PHILOSOPHY

APPROVED, THESIS COMMITTEE:

[Signatures]

Richard G. Baraniuk, Chairman
Associate Professor of Electrical and Computer Engineering

[Signatures]

Don Johnson
Professor of Electrical and Computer Engineering and Statistics

[Signatures]

David Scott
Professor of Statistics

Houston, Texas

May, 1999
TRANSFORM-DOMAIN MODELING OF NONGAUSSIAN AND $\frac{1}{f}$ PROCESSES

Matthew S. Crouse

Thesis: DOCTOR OF PHILOSOPHY
Electrical and Computer Engineering
Rice University, Houston, Texas (May 1999)
TRANSFORM-DOMAIN MODELING OF NONGAUSSIAN AND $1/f$ PROCESSES

Matthew S. Crouse

Abstract

Classical Gaussian, Markov, and Poisson models have played a vital role in the remarkable success of statistical signal processing. However, a host of signals — images, network traffic, financial times series, seismic measurements, wind turbulence, and others — exhibit properties beyond the scope of classical models, properties that are crucial to analysis and processing of these signals. These properties include a heavy-tailed marginal probability distribution, a nonlinear dependency structure, and a slowly-decaying or nonstationary correlation function. Fourier, wavelet, and related transforms have demonstrated a remarkable ability to decorrelate and simplify signals with these properties. Although useful transform-domain algorithms have been developed for signal analysis and processing, realistic transform-domain statistical models have not.

In this thesis, we develop several new statistical models for signals in the transform-domain with an eye towards developing improved algorithms for tasks such as noise removal, synthesis, classification, segmentation, and compression. We primarily focus on the wavelet transform, with its efficient multiresolution tree structure, and the Fourier transform. However, the theory, which is rooted in topics such as probabilistic graphs, hidden Markov models, and fractals, can be applied in a much more general setting. Our models have led to new algorithms for signal estimation, segmentation, and synthesis as well as to new insights into the behavior of data network traffic, insights potentially useful for network design and control.
To the memory of my father, Richard L. Crouse
Acknowledgments

I would like to give a big "hi-five" to Roger Claypoole, Tim Dorney, Felix Fernandes, Rohit Gaikwad, James Lewis, Ramesh (Neelsh) Neelamani, Robert Nowak, Jan Odegard, Vinay Ribeiro, Justin Romberg, and the entire Rice DSP group, past and present, for their insights and comradery.

I would also like to express my gratitude to my advisor Dr. Richard Baraniuk for fostering such an open, fun, and creative work environment. His commitment to excellence and hands-on mentoring have been invaluable, and I am honored to be the first among what will be a great line of his doctoral students.

I would like to thank the National Science Foundation (grant no. MIP–9457438) and the Office of Naval Research (grant no. N00014–95–1–0849) for their generous financial support. I would like to thank Rudolf Riedi for sharing with me his world-class expertise in fractals and multifractals, as well as thesis committee members Dr. Don H. Johnson and Dr. David W. Scott for providing useful commentary and criticism.

Finally, I must especially thank Thida Lwin for providing me with patient support and strong encouragement throughout my graduate studies.
Contents

Abstract iii
Acknowledgments iv
List of Illustrations x
List of Tables xiii

1 Introduction 1
1.1 Motivation Behind Transform-Domain Processing 1
1.2 Primary And Secondary Objectives 2
1.3 Three Challenges of Transform-Domain Modeling 4
1.4 Thesis Organization 7

2 Background 9
2.1 1/f Processes 9
  2.1.1 Continuous-time 1/f processes 9
  2.1.2 Discrete-time properties 13
  2.1.3 Discrete-time synthesis algorithms for Gaussian 1/f noise 17
2.2 The Discrete Wavelet Transform 19
  2.2.1 Definition 19
  2.2.2 Why the DWT is effective for signal processing 21
  2.2.3 The Haar wavelet transform and positive data 25
2.3 Probabilistic Graphs 27
2.4 Fractals and Multifractals 29
  2.4.1 Fractals 29
  2.4.2 Multifractals 30
3 Wavelet-Domain Hidden Markov Models For Statistical Signal Processing

3.1 Wavelet Trees ........................................... 37
3.2 Residual Properties of Wavelet Coefficients ......................... 38
3.3 Modeling Framework ..................................... 39
3.4 Wavelet-Domain Probability Models .............................. 40
  3.4.1 Probabilistic models for a single wavelet coefficient .......... 40
  3.4.2 Probabilistic models for a wavelet transform ................. 43
  3.4.3 Graph models for wavelet transforms ....................... 44
  3.4.4 HMT model .......................................... 45
  3.4.5 Extension to multi-dimensional data ....................... 47
  3.4.6 Three standard problems of HMMs .......................... 47
3.5 Training and Likelihood via the EM Algorithm ....................... 48
  3.5.1 EM algorithms for training ................................ 49
  3.5.2 Likelihood determination .................................. 50
  3.5.3 Robust training via tying .................................. 50
3.6 Applications ............................................. 51
  3.6.1 Wavelet-based estimation framework ....................... 52
  3.6.2 Estimation using wavelet-domain HMMs ..................... 53
  3.6.3 Wavelet-based detection and classification framework ..... 58
  3.6.4 Classification and detection of nonlinearity ............... 59
  3.6.5 Detection of an abrupt change ............................. 62
3.7 Persistence and An Upwards HMT ................................ 64
3.8 Discussion .............................................. 66

4 Context-Based Hidden Markov Models For Statistical Signal Processing

4.1 Definition and Use of Contexts ................................ 68
5.2 EM Algorithm for CHMMs ..................................................... 70
4.3 Applications ................................................................. 70
  4.3.1 Signal estimation ..................................................... 71
  4.3.2 Signal synthesis ..................................................... 73
4.4 Discussion ................................................................. 76

5 The Multifractal Wavelet Model With Application To Network Traffic

5.1 Notation ................................................................. 78
5.2 Multifractal Wavelet Model ............................................. 79
  5.2.1 Positivity through multiplication .................................. 79
  5.2.2 Synthesis procedure ............................................. 79
  5.2.3 Closed-form coefficient expressions ............................. 80
  5.2.4 Properties of the MWM ......................................... 81
  5.2.5 The MWM is a multifractal .................................... 84
  5.2.6 Related work ..................................................... 84
5.3 Data Modeling Using the MWM ......................................... 85
  5.3.1 Controlling the wavelet energy decay ....................... 85
  5.3.2 Controlling the moments of the scaling coefficients ....... 87
  5.3.3 Distributions for the multipliers ................................ 87
  5.3.4 Distribution for the root scaling coefficient ............... 89
  5.3.5 Modeling positive 1/f noise .................................. 89
  5.3.6 Fitting the MWM to data measurements ...................... 91
5.4 Application to Network Traffic ...................................... 92
  5.4.1 Modeling interarrival times ................................... 92
  5.4.2 Modeling packets-per-time ................................... 97
  5.4.3 Queuing behavior ............................................... 100
5.5 Discussion ............................................................. 102
6 FFT-Based Synthesis Of $1/f$ And Long-Range Dependent Data 104

6.1 Basic Approach ........................................ 105
   6.1.1 FFT properties and application ......................... 105
   6.1.2 Algorithm ........................................... 107
   6.1.3 Sufficient conditions for exactness ..................... 108
   6.1.4 Proof of exactness for $1/f$ Processes .................. 109

6.2 Exactness for Two Classes of Gaussian LRD Processes ............. 111
   6.2.1 Asymptotic discrete fractional Gaussian noise .......... 111
   6.2.2 Kinked fractional Brownian motion ..................... 112
   6.2.3 Gaussian LRD synthesis example ....................... 118

6.3 Synthesis of NonGaussian LRD Processes ........................ 120
   6.3.1 Creating a nonGaussian process from one Gaussian process .... 120
   6.3.2 Creating a nonGaussian process from two Gaussian processes .... 122
   6.3.3 Algorithm for synthesis of nonGaussian LRD processes ....... 123
   6.3.4 Example densities .................................. 124
   6.3.5 NonGaussian synthesis examples ....................... 127

6.4 Discussion ............................................ 129

7 Discussion And Future Work .................................. 132

7.1 Contributions of This Thesis ................................ 132

7.2 Some Open Issues ...................................... 134

A Appendix — EM Algorithm For Hidden Markov Trees 137

A.1 Set-up ................................................. 139

A.2 E step for a single wavelet tree (upward-downward algorithm) .... 140

A.3 E step for multiple wavelet trees .......................... 142

A.4 M step ............................................... 142

A.5 Tying within trees ...................................... 143
B Exactness Of FFT-Based Synthesis Of DfGn For $1/2 < H < 1$ 145

B.1 Case I: $n \geq 1$ ................................................. 145
B.2 Case II: $n = 0$ ................................................... 147

C Exactness Of FFT-Based Synthesis Of NonGaussian ESS

Processes For $1/2 < H < 1$ 148

C.1 Case I: $n \geq 2$ ................................................. 148
  C.1.1 General bound ............................................. 148
  C.1.2 Lognormal density ....................................... 150
  C.1.3 Uniform density .......................................... 150
  C.1.4 Exponential density ..................................... 151
  C.1.5 Pareto density: $\alpha \geq 3$ .............................. 151
C.2 Case II: $n = 0, 1$ ......................................... 152

Bibliography 155
Illustrations

1.1 Time-frequency tilings corresponding to time-domain, frequency-domain, and wavelet-domain signal representations. ........................................ 5

2.1 Tiling of the time-frequency plane by the atoms of the wavelet transform. .... 20
2.2 Haar scaling and wavelet functions and a binary tree of scaling coefficients. 26
2.3 Undirected and directed probabilistic graphs. .................................. 28
2.4 Deterministic and random fractals. .............................................. 30
2.5 A binomial multifractal, its integral, and its Hausdorff spectrum. ............ 31
2.6 Iterative construction of the binomial cascade. ................................. 34

3.1 Organization of a wavelet transform as a forest of binary trees. ............... 37
3.2 Examples of the Clustering and Persistence properties in the
time-frequency plane. ................................................................. 39
3.3 Gaussian mixture model for a single wavelet coefficient. ..................... 41
3.4 Fit of a Gaussian mixture model to real data. ................................ 42
3.5 Hidden Markov models for the wavelet transform. ............................ 45
3.6 HMT for an image quadtree. ..................................................... 48
3.7 Tying in the HMT model. .......................................................... 51
3.8 Block diagrams for wavelet-based denoising. ................................ 52
3.9 Denoising the Dopplet test signal in white Gaussian noise. .................. 56
3.10 Typical autoregressive signals used in nonlinear classification experiment. 60
3.11 Probability of error rates for classification of nonlinear AR data. .......... 61
3.12 Typical signals for the abrupt change detection experiment. 62
3.13 Probability of error rates for detection of an abrupt change. 63
3.14 Directed graph corresponding to an upwards HMT model. 64

4.1 Contextual models for the DWT. 69
4.2 Bellcore interarrival times and a synthetic CHMM trace. 75
4.3 Histograms of the interarrival times in Figure 4.2. 75

5.1 The MWM construction. 80
5.2 Plot of the density functions corresponding to several different values of the beta parameter $p$. 88
5.3 Interarrival times of Bellcore, August 1989 data, MWM data, and fGn data. 93
5.4 Variance-time plots for the Bellcore and MWM data. 94
5.5 Log-log plots for the scaling of the moments of the Bellcore data. 94
5.6 Multifractal spectra of the Bellcore and MWM data. 95
5.7 Log-log plots for the scaling of the moments of the MWM data. 97
5.8 Bytes-per-time traces corresponding to data from Lawrence Berkeley Laboratory, MWM data, and WIG data. 99
5.9 Histograms of the bytes-per-time traces in Figure 5.8. 100
5.10 Variance-time plots of the Berkeley, MWM, and WIG data. 101

6.1 Covariance function of kinked discrete fractional Gaussian noise without a transition. 114
6.2 Covariance function of kinked discrete fractional Gaussian noise with a linear transition. 116
6.3 Gaussian LRD synthesis example 119
6.4 Block diagram for the synthesis procedure for nonGaussian LRD processes. 121
6.5 NonGaussian synthesis example. 128
6.6 Example of the prewarping used to synthesize nonGaussian LRD processes. 129

C.1 Verifying the convexity of underlying Gaussian covariance functions required for the synthesis of nonGaussian ESS processes. . . . . . . . . . . . 153
Tables

3.1 Denoising results for several test signals and estimation techniques. . . . . 57
3.2 Empirical calculations regarding the inter-dependency of wavelet
coefficients in different test signals. . . . . . . . . . . . . . . . . . . . . . . . . 66

5.1 Asymptotic values for the beta parameter $p$ needed to fit positive ESS data
with Hurst parameter $H$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
Chapter 1

Introduction

1.1 Motivation Behind Transform-Domain Processing

The foundations of statistical digital signal processing (DSP) have been built on classical Gaussian, Markov, and Poisson models. Classical tools such as matched filters, autoregressive (AR) models, and Kalman and Wiener filters have become ubiquitous in applications ranging from digital communication to radar to speech coding and recognition [50, 83, 104]. But not all signals are well-approximated by classical models. Digitized images [103], network traffic loads [54, 87], geophysical data [39, 66], financial data [62, 67], and a host of other signals [32, 34] can exhibit properties outside the realm of a standard analysis. These properties can include a skewed or heavy-tailed marginal distribution, a nonlinear dependency structure, a slowly-decaying correlation function, nonstationarity, and a power spectrum of the form $1/|f|^\alpha$, $-1 < \alpha < 3$.

In conjunction these properties can be extremely difficult to accurately measure and model within a comprehensive theoretical framework. For instance, modeling Gaussian signals that have a power spectra of the form $1/|f|^\alpha$ has been fraught with difficulty (except for the two notable exceptions of $\alpha = 0$ and $\alpha = 2$). These signals can be long-range dependent (LRD), exhibiting an extremely slowly-decaying covariance function that is impossible to fit accurately using classical AR or Markov models. Add an extra degree of complexity by considering highly nonGaussian $1/f$-type signals, and we enter a realm where classical models break down.

Undaunted by the task of modeling such processes, we look to the field of data compression, where transform coding has demonstrated a remarkable success with these types of signals [44, 81, 95]. To transform-encode a signal, we apply a (typically linear) trans-
formation and then quantize and entropy-encode the resulting transform coefficients. To reconstruct the signal, we entropy-decode the transform coefficients and invert the transform. The point behind such coding is to use a transform that compacts a signal into a handful of non-zero coefficients that can be encoded easily and efficiently. Indeed, in digital image compression we observe that the discrete wavelet transform (DWT) combined with a simple coding scheme [95] has demolished the performance of coders that operate directly on the image (e.g., Differential Pulse Code Modulation [44]).

We wish to leverage the results from compression into a more general analysis framework by understanding the reason why transform coding has been so successful. One explanation is that transforms strip away much of the "structure" in the data. That is, the transform converts signals into data that is much easier to model and, hence, encode in a near-optimal fashion. Solid theoretical results support this belief, demonstrating that wavelet and related transforms serve as unconditional bases for several signal classes and hence are in some sense "optimal" for representing complicated signals [28]. Since transforms allow us to model signals more simply and more accurately, we can develop powerful new algorithms for a host of applications including, but not limited to, data compression. The claim that compression, classification, and signal estimation are equivalent, though an oversimplification, rings true to the extent that the simpler and the more accurate the underlying model, the better the resulting algorithm. To date, most of compression has focused on efficient data structures, whereas for general problems we are interested in accurate probability models.

1.2 Primary And Secondary Objectives

Our first primary objective is to develop fast and efficient wavelet-domain probability models that are rich and flexible enough to capture the structure of a variety of complicated "real-world" signals. The definition of "real world" will remain somewhat nebulous, but includes signals that are smooth with occasional singularities (e.g., edges, ridges), signals for which classical models are ineffective. Digital photos serve as a canonical example.
Our second primary objective is to develop fast, efficient, and accurate transform-domain models for $1/f$ and LRD signals. Though they span a number of different fields, these processes are poorly understood, lying outside the confines of classical modeling techniques. Although our $1/f$ and LRD models can be applied to a variety of problems, we will focus on data networking, since it is becoming increasingly vital, and it conjures up many of the salient issues that occur in modeling these processes.

Our third primary objective is to develop a specific set of tools useful for data networking and similar applications where nonGaussian and $1/f$ processes are prevalent. Recently, traffic loads and interarrival times in data networks have been shown to exhibit LRD and self-similar behavior [54, 105]. This behavior, ill-described using classical Poisson or Markov models, has proven to be a key factor driving network performance. For a given mean traffic load these Poisson and Markov models predict levels of network performance (in terms of queue size, average delay, and packet loss) that are much more optimistic than what is observed in practice [30, 79]. Gaussian $1/f$ models such as fractional Brownian motion (fBm) and its increments process fractional Gaussian noise (fGn) have led to more realistic performance predictions [73], since the models characterize the traffic’s LRD — the fact that traffic loads exhibit strong correlations over long time periods. LRD in traffic leads to poor network performance because packets can arrive in great bursts over long time scales (over-utilization) followed by long periods of relative quiet (under-utilization).

Several questions of vital importance remain regarding the effect of traffic characteristics on network performance. What are the differing impacts of LRD and short-term correlations on network performance? More specifically, is there a time scale after which LRD becomes unimportant [71]? Are queueing results derived from inherently Gaussian fBm and fGn models realistic considering that traffic can be highly nonGaussian?

To begin to address these and similar questions, we pose five secondary objectives of this thesis:

1. Develop models to synthesize high-fidelity artificial traffic for use in realistic network simulations.
2. Create synthesis tools to determine, via simulation, (a) the differing complementary effects of short-term correlations and LRD in the queue, and (b) the effects of non-Gaussianity on the queueing behavior of LRD traffic.

3. Construct realistic models to help determine, via analysis, the same two questions in objective 2.

4. Develop tools to ascertain the "state of the network" (as measured through parameters of a probability model) for use in network control algorithms.

5. Develop techniques to validate the presence and significance of multifractality in network traffic.

Let us emphasize that we do not intend to solve these problems outright, but rather to provide DSP tools to help solve them. See [86] for work that has begun applying our tools to solve these networking problems.

We explicitly spell out these secondary objectives to help the reader understand some of the subtle differences between our $1/f$ and LRD noise models. These objectives will also help motivate some of the new models and techniques that we develop. However, the application of these tools does not solely lie in networking. We will also see that these tools developed here have great potential to be used in a much wider setting. For instance, these tools could be applied to characterize the volatility in the financial markets [10, 62]. Volatility exhibits non-Gaussian and LRD properties and is of great interest in the area of risk management. Similarly, the new tools we develop are potentially well-suited for modeling fractal scaling processes that occur in natural phenomena [5].

1.3 Three Challenges of Transform-Domain Modeling

Although a transform can simplify the problem of signal modeling, it does not solve it. Indeed, it poses new and intriguing challenges that we must address. We face three fundamental challenges in the quest to accurately and efficiently model and process signals in the
Figure 1.1: Example basis functions and time-frequency plane tiling for the time-domain, Fourier-domain, and wavelet-domain representations. In the different representations, each basis function may be viewed as occupying a rectangular region of the time-frequency plane. Roughly speaking, the transform coefficients (represented by •'s) summarize the signal content in each region. For many signals of interest, the statistical dependencies are strongest between coefficients that are close together in the time-frequency plane. We seek to incorporate these dependencies into our transform-domain models, observing that, although the time- or frequency-domain coefficients can be organized naturally into a 1-d series, the wavelet-domain coefficients cannot. The same is true for the coefficients of wavelet-packet, best-basis, and related adaptive transforms [107].

transform domain. To begin with, the transformed signal exhibits a dependency structure that is oftentimes incompatible with standard time-series models. To better understand this phenomena, let us consider the time-frequency perspective as shown in Figure 1.1. For many transforms of interest, each basis function may be viewed as containing information about signal content at a localized set of times and frequencies. Its associated transform coefficient roughly summarizes the signal content at that region of the time-frequency plane.

For most signals, the dependency between transform coefficients tends to be relatively localized in the two-dimensional time-frequency plane [38, 48, 61, 75]. For instance, those basis functions that are distant in time-frequency tend to have corresponding coefficients that are almost independent — for many natural signals, the signal content at time $t_1$ and frequency $f_1$ will generally be unrelated to the content at time $t_2 \gg t_1$ and frequency
$f_2 \gg f_1$.

For the wavelet and related transforms, arranging the coefficients into a 1-d series hides much of the local dependency structure in the time-frequency tiling. However, if we tailor our data structure to the transform, we can potentially describe these dependencies in an accurate, concise manner. For these signals, a new topology is imperative.

Aside from the new dependency structures elicited by a transform, the marginal probability density (or densities) of the transform coefficients present a modeling challenge. For DSP applications in general, the probability model of choice has been the Gaussian, since it leads to analytically tractable and intuitively pleasing DSP algorithms.

Unfortunately, the transform can make non-Gaussian data appear to be even more non-Gaussian. Recall that a "good" transform compacts most of the signal information into a few large non-zero coefficients, with the remaining coefficients close to zero. Assuming the variance of the transform coefficients is roughly equal to that of the original signal,\(^1\) then the overall distribution of transform coefficients will be more "peaky" at zero and "heavy-tailed" than that of the original signal; perusing a raw histogram of the transform coefficients, we will find that these coefficients appear to be less "Gaussian-like" than the signal itself. In some cases it can be argued (and rightly so if the signal is Gaussian) that the transform coefficients are Gaussian, but with differing variances. However, if these variances are not known a priori, than the overall behavior is indistinguishable from non-Gaussian and must be accounted for in any accurate model.

Lastly, we point out that many simple time-domain or spatial-domain signal properties are difficult to account for in the transform domain. For example, if we wish to obtain a desired marginal distribution for the signal, then it is almost impossible to derive a tractable joint probability distribution for the transform coefficients (except in the case of a Gaussian signal). More specifically, determining the constraints on the Fourier transform coefficients to assure the positivity of a signal is cumbersome, let alone formulating a reasonable joint probability distribution for these coefficients. For the wavelet transform, the second-

\(^1\)This assumption is reasonable since many transforms are orthogonal or approximately so.
order stationarity of a time-domain signal results in a nontrivial covariance structure of the wavelet coefficients.

Thus, while transform-based signal modeling and processing is promising, to do it well we must surmount three potential obstacles: the necessity for new data structures, the potential nonGaussian attributes of the transform coefficients, and the difficulty in the transform-domain modeling of time-domain properties.

1.4 Thesis Organization

This thesis covers a diverse set of topics. Chapter 2 will be devoted to notation and necessary introductory material in $1/f$ processes, wavelet theory, fractals and multifractals, and probabilistic graph theory.

Targeting our first primary objective, in Chapter 3 we will introduce wavelet-domain Hidden Markov Models, a class of models that can be applied effectively for general-purpose signal processing algorithms. These models will specifically address the problem of developing efficient data structures to model nonGaussian wavelet coefficients. The notion of context-based Hidden Markov Models in Chapter 4 will serve as a useful variation on the Hidden Markov Model theme. Although the focus will be on wavelet transforms, the overarching framework developed here can be applied to other transforms as well.

Targeting our second and third primary objectives, in Chapter 5 we introduce the Multifractal Wavelet Model to the difficult problem of accurately, yet succinctly, modeling and synthesizing positive $1/f$ data. Model in hand, we look at a specific application to modeling and synthesis of network traffic.

Despite the power and flexibility of the Multifractal Wavelet Model, we will find that it cannot exactly model LRD processes. One question naturally arises: How do we exactly and efficiently model LRD and $1/f$ type processes? In Chapter 6, we will prove that the Fast Fourier Transform (FFT) can exactly (i.e., matching second-order statistics and first-order marginal density) synthesize a wide class of Gaussian and nonGaussian processes. For exact synthesis of the ever-popular fBm and fGn processes, this algorithm far
outstrips competing algorithms in terms of computational complexity \(O(N \log N)\) versus \(O(N^2)\)). Leveraging the power of the FFT approach, we will develop some intriguing generalizations to the powerful fBm and fGn models. In Chapter 7, we will summarize the contributions of this thesis and discuss some problems of future interest.
Chapter 2

Background

In this chapter we will review some basic properties of the $1/f$ and LRD processes that we are trying to model. We will also provide a whirlwind review of wavelets, with a brief discussion of why wavelets are useful for signal processing. We will then discuss fractals and multifractals, relating them to $1/f$ noise and our wavelet-domain modeling. Finally, we will review some theory of probabilistic graphs, which will be crucial for later development of wavelet-domain probability models.

2.1 1/f Processes

$1/f$, self-similar, and long-range dependent (LRD) processes occur frequently in numerous disciplines, including physics, chemistry, astronomy, economics, and biology [34, 51, 109]. In electrical engineering, these processes have served as extremely useful models for characterizing textures in images [43, 46, 58] and noise in analog circuits [51]. In finance, these processes have been recognized as important for characterizing price volatility [10, 62]. They have even had an impact on DNA research [4]. We find the study of $1/f$ processes rewarding because of their widespread applicability, their fascinating and beautiful properties such as self-similarity, and their powerful impact in key fields of interest.

2.1.1 Continuous-time 1/f processes

We now provide a brief review of some of the properties of continuous-time and discrete-time $1/f$ processes, with a focus on fBm and fGn. Although a continuous-time treatment is key to understanding and analyzing the basic properties of $1/f$ processes, in the end a discrete-time treatment is needed to accurately analyze and synthesize these processes.
using the digital computer.

Consider a second-order stationary Gaussian process $X(t)$ with covariance function

$$r_X(\tau) \equiv \mathbb{E}[X(t)X(t-\tau)] - \mathbb{E}[X(t)]^2,$$  \hfill (2.1)

and Fourier power spectrum

$$S_X(f) \equiv \int_{-\infty}^{\infty} r_X(\tau) e^{-j2\pi f \tau} d\tau, \quad -\infty < f < \infty.$$  \hfill (2.2)

Such a process is called a continuous-time $1/f$ process if it has a Fourier power spectrum of the form [111]

$$S_X(f) \propto \frac{1}{|f|^\alpha}, \quad -\infty < f < \infty, \quad -1 < \alpha < 3.$$  \hfill (2.3)

Right at the outset we encounter some technical difficulties. For the case $1 < \alpha < 3$, $S_X(f)$ is not integrable near the origin and hence, strictly speaking, not a valid power spectrum. This is usually attributed to an inherent nonstationarity in the process, in which case (2.1) and (2.2) are not valid definitions. In this case, (2.3) must be interpreted as a time-averaged power spectrum [37].

When $-1 < \alpha < 1$, $S(f)$ is not integrable as $f \to \infty$, since the high-frequency energy decays too slowly. Thus, $1/f$ processes for $-1 < \alpha < 1$ have infinite power and are well defined only in a generalized sense.¹ Nevertheless, these processes can still provide extremely useful models. For instance, continuous-time white Gaussian noise, $1/f$ noise with $\alpha = 0$, is ubiquitous in communication theory as a model for channel noise. Let us meet the Gaussian $1/f$ processes for $\alpha \neq 0$.

**Fractional Brownian motion (fBm):** $1 < \alpha < 3$

The Gaussian process used for modeling $1/f$ noise with $1 < \alpha < 3$ is known as fractional Brownian motion (fBm) [63]. The fBm process $B_H(t)$ is a continuous, jointly Gaussian

---

¹See [111] for one such definition that, in addition, treats the case of $1/f$ processes for $\alpha \leq -1$ and $\alpha \geq 3$. 
process defined via the self-similarity property

\[ B_H(at) \overset{\text{d}}{=} a^H B_H(t), \] (2.4)

with \( 0 < H < 1 \) denoting the Hurst parameter and the notation \( \overset{\text{d}}{=} \) representing equality in finite-dimensional distributions. In words, a zoom of fBm by a factor of \( a \) leads to an fBm with statistical properties that are identical, after scaling by \( a^H \), to the original fBm.

For fBm the time-averaged power spectrum is given by (2.3) with \( \alpha = 2H + 1 \) and the nonstationary covariance function by [63]

\[ r_{B_H}(s,t) \equiv \mathbb{E}[B_H(t)B_H(s)] = \frac{\sigma^2}{2} \left[ |s|^{2H} + |t|^{2H} - |s-t|^{2H} \right]. \] (2.5)

**Fractional Gaussian noise (fGn):** \( -1 < \alpha < 1 \)

The Gaussian 1/f process with \( -1 < \alpha < 1 \) can be viewed as a derivative of fBm with \( 1 < \alpha < 3 \), since a derivative provides a multiplicative factor of \( 1/f^2 \) in the power spectrum. In this case, the decay parameter \( \alpha \) relates to the Hurst parameter \( H \) via \( \alpha = 2H - 1 \). When \( 0 < \alpha < 1 \), using the generalized Fourier Transform inverse for \( 1/|f|^\alpha \) we obtain the stationary covariance function for the idealized derivative of fBm:

\[ r(\tau) \propto \tau^{\alpha-1}, \quad 0 < \alpha < 1. \] (2.6)

As noted previously, for \( -1 < \alpha < 1 \), a process with strict power-law decay can be defined only in a generalized sense. Therefore, we find it useful to construct an approximation to 1/f noise, \( -1 < \alpha < 1 \), using the continuous-time increments process for fBm,

\[ X_H(t) \equiv B_H(t + T) - B_H(t), \] (2.7)

which is a stationary Gaussian process known as fractional Gaussian noise (fGn) [63]. It has a spectrum of the form [37]

\[ S_{X_H}(f) = c \frac{\sin^2(\pi f T)}{|f|^{2H+1}}, \] (2.8)
with $c$ a proportionality constant. For $f \ll \frac{1}{T}$, we have $S_{X_H}(f) \propto 1/|f|^{2H-1}$. The accompanying covariance function is given by

$$r_{X_H}(\tau) = \frac{\sigma_x^2 T^{2H}}{2} \left[ |\tau + 1|^{2H} + |\tau - 1|^{2H} - 2|\tau|^{2H} \right], \quad 0 < H < 1.$$  \hfill (2.9)

Note that $r_{X_H}(\tau) \simeq \tau^{2H-2}$ as $\tau \to \infty$, so fGn has the same asymptotic covariance decay as the idealized derivative of fBm.

**Long-range dependence**

For $1/2 < H < 1$, the slow power-law decay of $r_{X_H}(\tau)$ in (2.6) and (2.9) implies that the integral of the covariance function, $\int_0^{T_0} r_{X_H}(\tau) \, d\tau$ will diverge off to $\infty$ as $T_0 \to \infty$. For any stationary process $X$, this divergence in the integral of $r_X$ is known as long-range dependence (LRD) [17]. LRD is equivalent to a singularity in the DC component of the spectrum, $S(0) = \infty$, and is a key feature of fGn with $1/2 < H < 1$.

Since the covariance function of an LRD process decays slowly, LRD processes have a great deal of "memory." This is a key property in scenarios where LRD occurs, such as in data networking and the financial markets. In networking, LRD in traffic loads can lead to poor network performance, since the packets arrive in bursts over long time scales [30]. In financial markets, the memory aspect of LRD is potentially useful for predicting future volatility [10].

LRD is difficult to characterize using classical models such as AR or Markov models, since classical models exhibit a much faster exponential decay, and robust model parameter estimation for highly correlated data is difficult. For instance, approximating an LRD $1/f$ process using an autoregressive moving average (ARMA) model is equivalent to fitting a slowly-decaying power-law covariance function using quickly decaying exponentials — an impossible task for a finite-order model. This is one reason why fGn and fBm have recently come to the fore as models for LRD processes.
2.1.2 Discrete-time properties

A discrete-time treatment is needed to accurately analyze and synthesize $1/f$ processes using the digital computer. To start, we will examine the second-order behavior of two discrete-time processes: the sampled fBm process $B_H(nT)$ and the fGn process (2.7) sampled with period $T = 2^L$ known as discrete fractional Gaussian noise (dfGn):

$$X_H^{(L)}[n] \equiv X(n2^L) = B((n+1)2^L) - B(n2^L), \quad n = 0, 1, \ldots \quad (2.10)$$

For clarity of presentation, we will replace $X_H^{(L)}[n]$ with $X_H[n]$ when convenient.

To model these processes, we will develop three equivalent second-order characterizations: the power spectrum, the covariance function, and the variance-time plot. An understanding of these different characterizations provides insight into different approaches for modeling and synthesizing LRD processes.

The power spectrum and why we should avoid it

In practice, we can only work with discrete-time finite-length data, so the Fourier power spectrum is of little practical use. A direct frequency-domain analysis or synthesis requires use of the Discrete Fourier Transform (DFT), which we will call the FFT for convenience.\(^3\)

The forward FFT can be written as

$$\lambda[k] \equiv \sum_{n=0}^{N-1} r[n] \exp\left(-j\frac{2\pi}{N}nk\right), \quad k = 0, 1, \ldots, N-1, \quad (2.11)$$

and the inverse FFT (IFFT) written as

$$r[n] \equiv \frac{1}{N} \sum_{k=0}^{N-1} \lambda[k] \exp\left(j\frac{2\pi}{N}nk\right), \quad n = 0, 1, \ldots, N-1. \quad (2.12)$$

\(^2\)We can sample with any $T > 0$, but will stick to $T = 2^L$, $L \in \mathbb{Z}$ to remain consistent with the notion of a signal approximation at dyadic wavelet scales.

\(^3\)The FFT is simply an efficient $O(N \log N)$ algorithm for computing an $N$-point DFT.
We define the FFT power spectrum (periodogram) of $X[n]$ as [50]

$$
\tilde{S}_X(f_k) \equiv \mathbb{E} \left[ \left| \frac{1}{N} \sum_{n=0}^{N-1} X[n] \exp(-j2\pi nf_k) \right|^2 \right], \quad f_k = \frac{k}{N}, \text{ for } k = 0, 1, \ldots, N-1.
$$

(2.13)

which is the expected value of the squared magnitude of the FFT of a length-$N$ realization of the process $X[n]$. The FFT power spectrum provides approximate values of the Discrete Time Fourier Transform (DTFT) power spectrum at frequencies $f = f_k$.

The FFT power spectrum of a length-$N$ dfGn sequence can be expressed in terms of the Fourier spectrum of fGn (2.8) as follows [50]

$$
\tilde{S}_{X_H}(f_k) = \left( \frac{\sin^2(\pi f N)}{N \sin^2(\pi f)} \otimes \frac{c}{T^2} \sum_{m=-\infty}^{\infty} \sin^2(\pi(f - m)) \left| \frac{f - m}{T} \right|^{-(2H+1)} \right),
$$

(2.14)

$$
f_k = \frac{k}{N}, \quad k = 0, 1, \ldots, N-1,
$$

with $\otimes$ the circular convolution operator. The infinite sum in (2.14) is caused by aliasing of higher frequencies onto lower frequencies (due to sampling), and the convolutional smearing is caused by the effects of finite-length data. This complicated expression deviates noticeably from a strict $1/f$ behavior and is difficult to calculate. Even worse, it behaves as a nontrivial function of the data length $N$.

Of course, the same type of sampling effects and smearing occurs in the FFT power spectrum of sampled fBm, except with an added difficulty in interpretation — fBm is non-stationary, and the notion of power spectrum applies only to stationary processes. Hence, the inherent difficulties in using and interpreting the FFT power spectrum suggest that we should look to alternative second-order characterizations of dfGn and sampled fBm.

**The covariance function and why we should use it**

Unlike the spectrum, which is warped and smeared by sampling, the covariance function for sampled fBm is obtained simply by plugging $s = nT$ and $t = mT$ into (2.5). Similarly, for dfGn the covariance function is found by substituting $nT$ for $\tau$ in (2.9). Moreover, we
can be confident that if we exactly match the covariance, then we have exactly matched all second-order statistics of the process, including the spectrum. In this paper, we will focus on modeling the covariance behavior of dfGn, since dfGn is easier to synthesize than sampled fBm and can be converted directly to sampled fBm.

Since the choice of the sampling period $T$ effects only a constant scaling factor in the covariance of dfGn, without loss of generality we can set $T = 1$ in (2.9) to obtain

$$r_{X_H}[n] = \frac{\sigma^2_X}{2} \left[ |n + 1|^{2H} + |n - 1|^{2H} - 2|n|^{2H} \right], \quad 0 < H < 1. \quad (2.15)$$

For $0 < H < 1/2$ and $n \neq 0$, the covariance function is negative. For $1/2 < H < 1$, the covariance function is positive and decreasing in $|n|$ with an asymptotic power-law decay $n^{2H-2}$ as $n \to \infty$. This decay implies dfGn exhibits LRD for $1/2 < H < 1$ since, in accordance with the continuous-time LRD definition, a discrete-time process $X$ is LRD if $\sum_{n=0}^{N} r_X[n]$ does not converge as $N \to \infty$.

Note that, strictly speaking, dfGn has a mean of zero. However, in networking and other applications, non-zero mean processes with the covariance structure of dfGn prove useful. In this paper, we will not discriminate between zero and non-zero mean dfGn.

**Variance-time plot and generalizations of fBm and fGn**

An alternative characterization of the second-order properties of fBm and fGn can be obtained using the self-similarity property of fBm (2.4)

$$\text{Var}[B_H(t + s) - B_H(t)] = |s|^{2H} \text{Var}[B_H(t + 1)] = |s|^{2H} \sigma_X^2$$

or equivalently for dfGn with $T = 1$

$$\text{Var} \left[ \sum_{n=n_0}^{n_0+m} X_H[n] \right] = m^{2H} \sigma_X^2$$

for $m \in \mathbb{N}$.
The latter property can be measured by forming the aggregated processes\(^4\)
\[
X_H^{(L+1-m)}[n] \equiv \frac{1}{m} \sum_{i=(n-1)m+1}^{nm} X_H[i],
\] (2.18)
and tracking how their variances decrease with the aggregation variable \(m\). It follows from (2.17) that
\[
\text{Var}\left[X_H^{(L+1-m)}[n]\right] = m^{2H-2} \sigma_X^2.
\] (2.19)
Hence, a log-log plot of the variance of \(X_H^{(L+1-m)}[n]\) as a function of \(m\) — known as a \textit{variance-time plot} — will have a slope \(2H - 2\). The variance-time plot can be applied to nonGaussian and non-zero-mean data as well. Processes whose variance scales with a power of \(2H - 2\) have been termed \textit{exactly second-order self-similar (ESS) processes} \cite{105}.\(^5\) Since ESS processes have power spectra identical to dfGn, a Gaussian 1/f noise, we can say that ESS processes correspond to nonGaussian 1/f noise. However, considering the difficulties associated with the FFT power spectrum (e.g., they are not truly 1/f), the terminology ESS provides a more precise characterization.

For more general processes, we can define a \textit{structure function} \(g_B\) \cite{48, 112} such that
\[
\text{Var}\left[B(t + s) - B(t)\right] = g_B(s) \sigma_X^2
\] (2.20)
and for \(T = 1\) (now in a discrete-time setting)
\[
\text{Var}\left[X^{(L+1-m)}[n]\right] = \frac{g_B[m]}{m^2} \text{Var}\left[X[n]\right].
\] (2.21)
When \(g_B(s) = \sigma_X^2 |s|^{2H}\), we have the self-similarity condition for fBm and fGn. The structure directly describes the scaling behavior of a given process and provides a more intuitive description of LRD behavior than the covariance function. However, the structure function converts directly to the covariance function via one-half the central second difference
\[
r_X[n] = \frac{1}{2} (g_B[n+1] - 2g_B[n] + g_B[n-1]).
\] (2.22)

\(^4\)Recall from (2.10) that we write \(X_H[m]\) instead of \(X_H^{(L)}[m]\), with \(T = 2^L\) our dfGn sampling rate.

\(^5\)Strictly speaking, the Hurst parameter \(H\) only applies to fBm and fGn. However, we will use it in the context of more general LRD data to characterize the covariance decay.
We will use the structure function in Section 6.2 to characterize more general types of LRD processes.

### 2.1.3 Discrete-time synthesis algorithms for Gaussian $1/f$ noise

We now describe several commonly used alternatives for synthesizing the Gaussian $1/f$ processes corresponding to sampled fBm and dfGn. It is simple to convert between these processes using (2.10), so a synthesis algorithm for one can synthesize the other as well.

**Cholesky factorization**

The method is exact [58, 101], but slow, requiring $O(N^2)$ computations to form a length-$N$ dfGN vector $X$. From (2.9), we form the Toeplitz covariance matrix $R_{XH}$ for dfGn and then factor it into $R_{XH} = QQ^T$ via the Levinson-Durbin algorithm (direct Cholesky factorization is slower — $O(N^3)$). We then form $X = QV$, with $V$ a white Gaussian noise vector, to obtain the desired dfGn process.

**FFT synthesis based on the power spectrum**

If one is to use the FFT power spectrum for synthesis, a reasonable approach is to ensure that the FFT power spectrum of the synthesized process has properties equivalent to those of the FFT power spectrum of the data to be modeled.

Paxson [78] takes this approach, performing an FFT-based synthesis of dfGn under the assumption that for a single realization the FFT power spectrum at $S_{XH}(f_k)$ is independent and exponentially distributed with mean given by the infinite sum in (2.14). They thus generate exponentially-distributed powers for each FFT frequency bin, multiply in a random phase, and invert to obtain the desired dfGn signal.

Although fast, Paxson's approach does not account for the effects of the convolutional smearing in (2.14) nor for the fact that the power in different FFT frequency bins can be correlated for dfGn. Furthermore, the output process has an undesirable circulant covariance structure (see Section 6.1), which among other things implies that the end of the
synthesized dfGn trace will often be highly correlated with the beginning.

In passing, we note that any direct FFT-based synthesis of sampled fBm through this approach is undesirable because the same sampling and smearing effects occur for sampled fBm. Moreover, any process synthesized by this FTT approach will be stationary, whereas fBm is nonstationary.

**Random midpoint displacement**

The next technique [52] synthesizes the values of a sampled fBm trace at a few dispersed points and then successively refines (or randomly interpolates) the values between the synthesized points. Since it does not attempt to model all cross-covariances between all fBm samples, it is inexact and in practice can deviate noticeably from exact fBm.

**Parametric approaches**

Corsini and Saletti [16] apply to white noise a discretized set of analog filters with logarithmically-spaced poles and zeros in order to generate a process with an approximate $1/|f|^\alpha$ power spectrum for $-2 < \alpha < 2$. Aside from the aliasing and finite-length concerns mentioned earlier, this approach is inexact due to the discretization of analog filters.

The fractionally-differenced autoregressive integrated moving average (FARIMA) model combines an ARMA model for modeling short-range dependence with a fractional differencing $d$ operation for describing long-range dependence [42,49]. Thus, the FARIMA model can approximate not only $1/f$ noise, but a wide class of LRD processes. However, estimation of ARIMA parameters can be difficult, and exact synthesis of ARIMA data is $O(N^2)$.

**Self-similar basis functions**

The remaining two techniques construct $1/f$ noise from self-similar basis functions that are logarithmically-spaced in frequency. The first uses wavelets and relies on the fact that the discrete wavelet transform (DWT) in $O(N)$ computations, provides an approximate
Karhunen-Loève (KL) transform for 1/f processes, including fBm and fGn [38, 47, 110, 111]. We defer an in-depth discussion of this approach to the review of wavelets in Section 2.2.2.

The randomized Weierstrass approach [32] synthesizes fBm by summing scaled sinusoidal functions (at logarithmically-spaced frequencies) with random height and phases. Again, the approach is inexact, although its deviation from true fBm has not been well quantified.

In Chapter 5, we will develop new wavelet models for positive-valued LRD and 1/f noise. In Chapter 6, we will develop a technique based on the FFT for exactly synthesizing fGn, fBm, nonGaussian ESS noise, and as a host of more general LRD processes.

2.2 The Discrete Wavelet Transform

2.2.1 Definition

The discrete wavelet transform (DWT) represents a 1-D real signal $X(t)^6$ in terms of shifted and dilated versions of a prototype bandpass wavelet function $\psi(t)$ and shifted versions of a lowpass scaling function $\phi(t)$ [21, 107]. For special choices of the wavelet and scaling functions, the atoms

$$
\psi_{j,k}(t) \equiv 2^{j/2} \psi(2^j t - k), \quad (2.23)
$$

$$
\phi_{j,k}(t) \equiv 2^{j/2} \phi(2^j t - k), \quad j, k \in \mathbb{Z} \quad (2.24)
$$

form an orthonormal basis, and we have the signal representation [21, 107]

$$
X(t) = \sum_{k} U_{j_0,k} \phi_{j_0,k}(t) + \sum_{j = j_0}^{\infty} \sum_{k} W_{j,k} \psi_{j,k}(t), \quad (2.25)
$$

---

$^6$We consider the signal $X(t)$ to be random and so use capital letters for all quantities derived from it.
with

\[ W_{j,k} \equiv \int X(t) \psi_{j,k}(t) \, dt, \]  
\[ U_{j,k} \equiv \int X(t) \phi_{j,k}(t) \, dt. \]  

Figure 2.1 provides a graphical depiction of the time-frequency analysis affected by the wavelet transform.

In this representation, \( j \) indexes the scale or resolution of analysis — larger \( j \) corresponds to higher resolution analysis. \( J_0 \) indicates the coarsest scale or lowest resolution of analysis. \( k \) indexes the spatial location of analysis.

For a wavelet \( \psi(t) \) centered at time zero and frequency \( f_0 \), the wavelet coefficient \( W_{j,k} \) measures the signal content around time \( 2^{-j}k \) and frequency \( 2^j f_0 \). The scaling coefficient \( U_{j,k} \) measures the local mean around time \( 2^{-j}k \). In the wavelet transform, \( j \) indexes the scale of analysis: \( J_0 \) indicates the coarsest scale or lowest resolution of analysis, and larger \( j \) correspond to higher resolutions of the analysis.

In practice, we only have access to a discrete-time finite-resolution representation of \( X(t) \). To handle this scenario, with loss of generality we will set \( J_0 = 0 \) and replace the semi-infinite sum in (2.25) with a sum over \( L \) scales \( 0 \leq j < L, \ L \in \mathbb{N} \). We then define the discrete-time approximation of the process \( X(t) \) at resolution \( L \):

\[ X^{(L)}[k] \equiv 2^{-L/2} U_{L,k} = 2^{-L/2} \int \phi_{L,k}(t) X(t) \, dt \]  

(2.28)
for $k = 0, \ldots, 2^L - 1$. $X^{(L)}[k]$ is the signal we actually use for calculating the DWT on a digital computer. When it is clear from the context, we will omit the superscript and write $X[k]$. Note that multi-dimensional wavelet bases for higher-dimensional data can be formed using tensor products of wavelet and scaling functions [21, 107].

2.2.2 Why the DWT is effective for signal processing

The wavelet transform has several attractive properties that make it natural for signal and image processing. We call these the primary properties of the wavelet transform:

Locality: Each wavelet atom $\psi_{j,k}$ is localized simultaneously in time and frequency.

Multiresolution: Wavelet atoms are compressed and dilated to analyze at a nested set of scales.

Compression: The wavelet transforms of real-world signals tend to be sparse.

Together the Locality and Multiresolution properties enable the wavelet transform to efficiently match a wide range of signal characteristics, from high-frequency transients and edges to slowly-varying harmonics. The wavelet transform's ability to match a wide variety of signals leads to the Compression property. Complicated signals can often be represented using only a handful of wavelet and scaling functions. As a result of these properties, statistical signal modeling and processing methods based in the wavelet-domain are, in many cases, much more effective than classical time-domain or frequency-domain approaches.

We will now examine the Compression property with more precision, developing some mathematical intuition behind the success of wavelets for signal processing. Classes of signals for which the wavelet transform is especially effective include $1/f$-type signals processes and piecewise smooth signals.

For Gaussian $1/f$ processes, the power of the wavelet transform is intimately linked with the notion of wavelets as an approximate Karhunen-Loève transform (KLT). For digital images and signals that belong to the class of piecewise smooth signals, the wavelet
transform has more to it than a KLT interpretation. In fact, the wavelet transform can have energy compaction properties that are superior to the KLT.

The DWT and 1/f processes

The inherent scaling property of the wavelet basis is well-suited for analyzing self-similar processes. Wavelets serve as an approximate KLT for 1/f processes [110], including fBm [38] and dfGn [47]. These highly correlated, LRD signals become nearly uncorrelated in the wavelet domain. This property has lead to the widespread use of wavelets for the analysis and synthesis of fractal and LRD signals [3].

In particular, the energy of the wavelet coefficients of a continuous fBm exhibits a power-law decay with scale [38]. The variance progression of the wavelet transform of sampled fBm and dfGn does not follow a strict power-law, but rather includes scale-dependent factors [38,47]. Kaplan and Kuo [47] have shown that for the Haar wavelet, the variance progression of the wavelet transform of dfGn satisfies

$$\text{var}(W_{j,k}) \propto 2^{-j(2H-1)}. \quad (2.29)$$

Moreover, the wavelet coefficients of dfGn are typically much less correlated than those of the underlying sampled fBm process. They use these facts to develop a robust wavelet-based estimator for the $H$ of a dfGn submerged in additive white Gaussian noise. Similar wavelet-based estimators for $H$ compare favorably with standard estimation techniques [2] and have been applied to practical problems such as network traffic analysis [3].

Note that all the decorrelation and energy decay properties of wavelet transforms hold for nonGaussian ESS processes (recall Section 2.1.2) just as they do for dfGn. The only difference is that the wavelet coefficients of nonGaussian ESS properties will have a non-Gaussian distribution.

Wavelets can also be used to synthesize approximate 1/f processes with generalized spectra of the form $\Gamma(f) \propto |f|^{-\gamma}$, $0 < \gamma < 2$, which includes fBm and fGn.\footnote{Processes corresponding to a wider range of $\gamma$'s can also be synthesized, using wavelets with regularity}
off the KLT property of the wavelet transform, Wornell generates zero-mean, independent Gaussian random variables \( W_{j,k} \) with power scaling according to [110]

\[
\text{var}(W_{j,k}) \propto 2^{-j\gamma}.
\]  

(2.30)

He then inverts the wavelet transform to obtain the synthesized process. Even though the mean and variance of the synthesized signal are stationary, this approach generally results in a nonstationary Gaussian process with time-varying correlation function (see Section 5.2.4). However, the time-averaged correlation and spectrum do approximate that of a 1/f process [110]. Though only approximate, this method's \( O(N) \) computational cost compares favorably with the \( O(N^2) \) cost of the Levinson algorithm for exact synthesis [101] and the \( O(N^3) \) cost of direct Cholesky factorization [58].

The DWT and Piecewise Smooth Signals

For digital images and signals that are piecewise smooth, wavelets provide extremely sparse representations that are asymptotically optimal in terms of nonlinear approximation error [15, 28]. One intuition behind this is as follows. By using a wavelet with \( R \) vanishing moments, the DWT can approximate polynomials up to order \( R \) using only scaling coefficients. For piecewise polynomial signals, the wavelet coefficients are zero except at the transitions between smooth regions. Moreover, at these transitions the number of nonzero coefficients decays rapidly as we go to finer scales due to the nature of increasingly localized wavelet basis functions. Hence, we expect the DWT to provide a parsimonious signal representation for these signals.

The optimality of wavelets derives from the fact that wavelets serve as unconditional bases for the function spaces containing singularity-rich data (e.g., data with edges, ridges), including the \( L^p \) \( (1 < p < \infty) \), Sobolev, Besov, and Triebel spaces [25, 28]. Unconditionality gives these bases optimal energy compaction properties: For signals in the above spaces, the expected nonlinear approximation error of a wavelet basis (the approximation
error as a function of the number of wavelet coefficients used) decays at an optimal rate [25, 28]. Another transform can exhibit a smaller error constant or a faster decay rate on a particular signal. However, over the entire class of signals no other transform can exhibit a faster decay rate.

To provide meaning to the above statements, let us briefly define these spaces of interest. Roughly speaking, the Besov space \( B_q^s(L_p) \) contains functions with \( s \) derivatives in \( L_p \) (the \( q \) parameter makes finer distinctions in smoothness). When \( s < 1 \), \( B_q^s(L_p) \) contains discontinuous functions (with edges). Interestingly, the Besov norm can be computed equivalently as a simple sequence norm on the wavelet coefficients [70]:

\[
\|X\|_{B_q^s(L_p)} \approx c_0 \|U_{j_0,k}\|_p + \left[ \sum_{j \geq j_0} 2^{js'q} \left( \sum_k |W_{j,k}|^p \right)^q \right]^{1/q}
\]  

(2.31)

with \( s' = s - 1/p + 1/2 \), \( q < \infty \), and \( c_0 = 2^{1/2-1/p} \). This norm is essentially an (un-normalized) higher-order sample moment (\( p \)-th within scale, weighted \( q \)-th across scale) of the wavelet coefficients. Sobolev spaces correspond to \( p = q = 2 \). Although the function-space description is not powerful enough to fully characterize real-world signals of interest, it does provide a key insight into their smoothness characteristics [25].

The DWT can be better than the KLT

Some researchers have explained the wavelet energy compaction phenomena is due to the fact that the wavelet transform is an approximate KLT. Although we have observed this is the case for Gaussian \( 1/f \) noise, for more general signals [15] points out a flaw in this argument, claiming that wavelets can compact energy even better than the true KLT of a process. The point is this: the KLT minimizes the linear approximation error, the difference between a signal and the approximation obtained by using the first \( N \) terms of the transform expansion. However, nonlinear approximation error involves calculating the difference between the signal and the \( N \) largest transform coefficients. In this case, the KLT is not necessarily the optimal linear transform. In fact, as has been observed in practice, the wavelet transform can be superior to the KLT in terms of approximation error decay.
rates [15, 28]. Moreover, the notion of nonlinear approximation is closer to the processing actually performed in top-of-the-line compression and estimation algorithms [29, 95]. For instance, in wavelet zero-tree coding, bits are allocated to a wavelet coefficient based on its size, not on its ordering in the expansion.

2.2.3 The Haar wavelet transform and positive data

The Haar wavelet transform will play a special role in this thesis. Not only is it the most easily understood wavelet transform, it has some unique properties useful for modeling nonnegative data.

In the Haar wavelet transform (see Figure 2.2), the prototype scaling and wavelet functions are given by

\[
\phi(t) = \begin{cases} 
1, & 0 \leq t < 1 \\
0, & \text{else}
\end{cases} \quad \text{and} \quad \psi(t) = \begin{cases} 
1, & 0 \leq t < 1/2 \\
-1, & 1/2 \leq t < 1 \\
0, & \text{else.}
\end{cases} \quad (2.32)
\]

The Haar scaling and wavelet functions provide the simplest example of an orthonormal wavelet basis. From (2.24) and (2.32) we see that the supports of the fine-scale scaling functions nest inside the supports of those at coarser scales; this can be neatly represented by the binary tree structure of Figure 2.2(b). Row (scale) \( j \) of this scaling coefficient tree contains an approximation to \( X(t) \) of resolution \( 2^{-j} \). Row \( j \) of the complementary wavelet coefficient tree (not shown) contains the details in scale \( j + 1 \) of the scaling coefficient tree that are suppressed in scale \( j \). In fact, the \( U_{j+1,k} \) consist simply of scaled sums and differences of the \( U_{j,k} \) and \( W_{j,k} \).

Wavelet-domain modeling of nonnegative processes is complicated by the fact that the wavelet coefficient constraints required to ensure a nonnegative output are nontrivial. Quite the contrary for the Haar wavelet, however. For the Haar wavelet, the scaling and wavelet
Figure 2.2: (a) The Haar scaling and wavelet functions \( \phi_{j,k}(t) \) and \( \psi_{j,k}(t) \). (b) Binary tree of scaling coefficients from coarse to fine scales.

Transform coefficients can be recursively computed using

\[
U_{j,k} = 2^{-1/2}(U_{j+1,2k} + U_{j+1,2k+1}) \quad \text{and} \quad (2.33)
\]

\[
W_{j,k} = 2^{-1/2}(U_{j+1,2k} - U_{j+1,2k+1}). \quad (2.34)
\]

We find from (2.32) and (2.27) that in the Haar transform each scaling coefficient \( U_{j,k} \) equals a scaled local mean of the signal. Hence, for nonnegative data we know that all \( U_{j,k} \geq 0 \). Rearranging (2.33) and (2.34) to

\[
U_{j+1,2k} = 2^{-1/2}(U_{j,k} + W_{j,k}) \quad \text{and}
\]

\[
U_{j+1,2k+1} = 2^{-1/2}(U_{j,k} - W_{j,k}), \quad (2.35)
\]

we thus find a simple constraint to guarantee that the process is nonnegative:

\[
|W_{j,k}| \leq U_{j,k}. \quad (2.36)
\]

Although we have derived (2.36) as a necessary condition, it is easy to see that it is also sufficient. This simple constraint will serve as the cornerstone of the MWM in Chapter 5. For more general wavelet systems (with longer, overlapping wavelets), the conditions are considerably more complex.
2.3 Probabilistic Graphs

Graphs and trees will play a central role in the development of hidden Markov models in Chapter 3. An undirected graph consists of a set of nodes labelled \( \{1, 2, \ldots, N\} \) and a set of connections linking the nodes. A path is a set of connections between two nodes. Two nodes are adjacent if there exists a path between them with no intervening nodes. A cycle is a "closed-loop," a path between a node and itself that does not traverse the same edge twice.

Probabilistic graphs are useful tools for representing the dependency relationships between a set of random variables \([56, 80, 97]\). These random variables could be discrete-valued, continuous-valued, or a combination of both. To construct a dependency structure for these random variables, we associate each random variable with a node in the graph; we model dependencies between variables by connecting the corresponding nodes.\(^8\) There exist two fundamental types of connections, directed connections and undirected connections, each with a different interpretation.

The designation for a directed connection in a probabilistic graph is "\( \rightarrow \)". \( W_1 \rightarrow W_2 \) means that the probability distribution for \( W_2 \) depends directly on \( W_1 \). \( W_1 \) is denoted the parent of \( W_2 \), and \( W_2 \) the child of \( W_1 \). The designation for undirected connection is "\( \leftrightarrow \)". In undirected case, the dependence relation is more "mutual" and better explained in the context of undirected graphs, graphs solely consisting of nodes and undirected edges.

In an undirected graph, a subset of nodes \( K \) separates two other subsets of nodes \( A \) and \( B \) if every path joining every pair of nodes \( i \in A \) and \( j \in B \) contains at least one node from \( K \). The notion of separation provides the probabilistic interpretation behind the undirected graph: If \( K \) separates \( A \) and \( B \), then given \( \{W_i\}_{k \in B} \), the sets of random variables \( \{W_i\}_{i \in A} \) and \( \{W_i\}_{j \in B} \) are independent. See Figure 2.3(a) for some concrete examples of this interpretation.

\(^8\)It is important not to confuse probabilistic graphs with state transition diagrams. In a probabilistic graph, each node corresponds to a different random variable. In a state transition diagram, each node corresponds to the value that a single random variable takes.
Figure 2.3: (a) An example of the dependency relationships in an undirected graph. \( W_1 \) is independent of all variables. \( W_4 \) is independent of all other variables given \( W_5 \). \( W_2 \) is independent of \( \{W_4, W_5\} \) given \( \{W_3, W_6\} \). (b) Similar-looking undirected and directed graphs can correspond to different probability structures. In the undirected graph on the left, \( W_2 \) and \( W_3 \) are dependent, but independent given \( W_1 \). In the directed graph on the right, \( W_2 \) and \( W_3 \) are independent, but dependent given \( W_1 \) — the opposite relationship! If the arrows in the directed graph were pointing downward, then the two graphs would be equivalent.

The general dependency relations in a directed graph, a graph solely consisting of nodes and directed edges, are not as simple to define. However, in the special case where no node has two or more nonadjacent parents, the directed graph has the same probabilistic properties as the undirected graph of the basic same structure.

A rooted tree is an undirected acyclic graph. In a tree there is a unique path linking any two nodes. All nodes that lie on the path from \( i \) to the root are called ancestors of \( i \); all nodes that lie on paths from \( i \) away from the root are called descendants of \( i \). The parent of \( i \) is its immediate ancestor and is denoted by \( \rho(i) \). A node is a child of \( i \) if \( i \) is its parent. We denote set of children of node \( i \) by \( \{j\}_{j \in ch(i)} \). A node may have several children, but only one parent; nodes with no children are called leaves of the tree. In a binary tree, each node that is not itself a leaf has two children.

Indices \( l(i) \) and \( r(i) \) represent, respectively, the adjacent indices to the left and right of \( i \); indices \( cl(i), cr(i) \), represent, respectively, the left and right children of \( i \).

Note that a directed graph is equivalent to an undirected tree if the all directions are from the root to the leaves. However, as Figure 2.3(b) shows, this is definitely not the case if the directions are from the leaves to the root. For undirected graphs that are trees, the
joint pdf has a very concise factorization. This property leads to efficient algorithms for parameter estimation and calculation of the joint pdf. Factorizations also exist for more general graphs, but the probability calculation and parameter become more complicated, particularly if the graph has cycles [56, 97].

2.4 Fractals and Multifractals

An extended discussion of fractals and multifractals is beyond the scope of this thesis. Nevertheless, a brief discussion is in order, since fBm is fractal, and the work in Chapter 5 has a strong connection with multifractals. For a more complete and mathematically rigorous discussion of multifractals, the interested reader is referred to [31–33, 89].

2.4.1 Fractals

Let us start with a definition of fractals. Speaking vaguely, we say that fractals are geometric objects exhibiting an intricate, highly irregular appearance on all resolutions [32, 65]. Another common feature of fractals is self-similarity — a zoomed version of a fractal will "look like" the original. The Sierpinsky triangle and fractional Brownian motion, examples of which are shown in Figure 2.4, provide good examples of self-similarity in deterministic and random fractals. Note that fBm is both a fractal and a $1/f$ process, hence fractality and $1/f$ properties are intertwined.

Mandelbrot has provided a definition that, while somewhat too strict, provides a precise mathematical characterization: A fractal is a set whose Hausdorff dimension is strictly greater than its topological dimension.

A point or disconnected set of points has a topological dimension zero, a curve has dimension one, and a surface dimension two. However, the Hausdorff dimension of sets of disconnected points, depending on their denseness and irregularity, can be greater than zero. Similarly, curves, depending on their denseness in the plane, can have a Hausdorff dimension greater than one. Such objects are fractal.
Figure 2.4: Examples of deterministic and random fractals: (a) the Sierpinsky triangle — any zoom-in of the triangle leads to an identical triangle and (b) a sample path of fractional Brownian motion \((H = 1/2)\) — any zoom-in of the process leads to a process with identical statistical properties.

To define Hausdorff dimension, we start with the definition of Hausdorff measure. For a set \(F\) in \(\mathbb{R}^N\), the Hausdorff measure is given by [32]

\[
H^s_\delta = \inf \left\{ \sum_{i=1}^{\infty} |B_i|^s : \{B_i\} \text{ is a cover of } F \text{ by balls of diameter at most } \delta \right\}, \\
H^s(F) = \lim_{\delta \to 0} H^s_\delta
\] (2.37)

with the \(|B_i|\) defined as the radius of the ball \(B_i\). Roughly speaking, the Hausdorff measure \(H^s(F)\) tells us the \(s\)-dimensional volume needed to cover a given set \(F\) using tiny balls. The Hausdorff dimension \(\dim_H(F)\) is derived from (2.37) as the value of \(s\) at which the Hausdorff measures \(H^s(F)\) jumps from \(\infty\) to 0.

2.4.2 Multifractals

Multifractals were first introduced to model dissipation of energy in turbulence [39, 64] and have proved well-suited to modeling non-homogeneous phenomena [67, 68]. More recently, the multifractal nature of network traffic has been demonstrated convincingly, first in [88] and subsequently in [35, 40]. The beauty of the multifractal formalism has motivated considerable research effort in mathematics [89]; however, few practical multifractal data models have been developed to date.
Figure 2.5: Example of (a) a binomial multifractal measure $C(t)$, (b) its integral $D(t)$, and (c) its Hausdorff spectrum $f_H(\alpha)$.

Just as for fractals, the exact definition of a multifractal is somewhat hazy. However, we can understand the essence of the term multifractal by studying specific properties of measures. Engineers typically use measures in the context of probability distributions that describe the random behavior of a signal, not as representations of the signal itself. However, any positive-valued signal itself may be viewed as a mass distribution or probability measure, or more precisely, as a finite-resolution approximation to a mass distribution.

A mass distribution $C(t)$ may be dispersed over a region in such a way that the concentration of mass varies widely, with the mass around any point $t$ decaying according to a power-law $\int_{t-\delta/2}^{t+\delta/2} C(t)\,dt \simeq \delta^{\alpha(t)}$ as $\delta \to 0$. If we collect all the points $t$ with the same decay rate into a set $K_\alpha = \{t : \alpha(t) = \alpha\}$, that set is typically fractal with a non-integer dimension. A mass distribution with this type of property is known as a multifractal measure, since each choice of $\alpha$ provides a different fractal set $K_\alpha$. For multifractal analysis the process $D(t) = \int_{\tau=0}^t C(\tau)\,d\tau$ is also of interest, since $C(t)$ may be well defined only in a generalized sense. See Figure 2.5 for an example of a multifractal measure and its integral.

One of the key tools of multifractal analysis is the multifractal spectrum. The (Hausdorff) multifractal spectrum $f_H(\alpha)$ is the plot of the fractal dimension of the sets $\dim_H K_\alpha$ versus $\alpha$. As shown in Figure 2.5(c), the spectrum $f_H(\alpha)$ is usually concave, taking values between zero and one. At $f_H(\alpha) = 1$, we have the local scaling behavior $\alpha(t)$ that occurs most frequently in the measure.
For interpreting the multifractal spectrum, the value of $\alpha(t)$ tells us the local properties about $t$ — the larger $\alpha(t)$ the smoother the signal is about $t$, while the smaller $\alpha(t)$ the spikier the signal is about $t$. The value of the spectrum $f_H(\alpha)$ roughly tells us "how many" points act like $\alpha(t) = \alpha$. Hence, a plot of the spectrum $f_H(\alpha)$ tells us the relative proportion of regions of smoothness and spikiness within $C(t)$. In a general sense the spectrum relates both to the nonGaussian marginal and the correlation structure of a signal.

**Multifractal Analysis Of Real Data**

One problem with the Hausdorff spectrum $f_H(\alpha)$ is that it cannot be calculated in practice. For real data we must develop other tools. For purposes of instruction, assume the measure $C(t)$ has support over the interval $[0, 1]$. In practice, we have access to only a finite-resolution representation of $C(t)$, such as on the dyadic scales

$$C^{(L)}[k] \equiv \int_{k2^{-L}}^{(k+1)2^{-L}} C(t)dt, \quad k = 0, 1, 2, \ldots, 2^L - 1$$

(2.38)

with $C^{(L)}[k]$ the discrete-time approximation to $C(t)$ at resolution level $L$. In our context, $C^{(L)}[k]$ is the actual signal we observe.

One property of interest involves the scaling of the $q$-th moment of the signal as a function of resolution level $L$:

$$T(q) \equiv \lim_{L \to \infty} \frac{-1}{L} \log_2 \mathbb{E} \sum_{k=0}^{2^L-1} (C^{(L)}(k))^q,$$

(2.39)

assuming that the limit exists. This partition function $T(q)$ can be estimated at a point $q = q_0$ by performing a linear fit of

$$E_j(q_0) \equiv \log_2 \sum_{k=0}^{2^j-1} (C^{(j)}(k))^{q_0}$$

(2.40)

versus $j$. (For $j$ large, we can remove the expectation operator of (2.39) under an ergodicity assumption on the data [89].)

It can be shown that the partition function relates to the Hausdorff spectrum via

$$f_H(\alpha) \leq T^*(\alpha) \equiv \inf_q (q\alpha - T(q)).$$

(2.41)
The relation $T^\ast(\alpha) \equiv \inf_q(q\alpha - T(q))$ is known as the Legendre transform of $T(q)$ [89]. For many signals of interest, (2.41) holds with equality [89]. Hence, a fitting of the log-log plots of the moments of a signal can be used to obtain a numerical estimate of its multifractal spectrum.

Neither the multifractal spectrum nor the partition function provides information about the exact moments or marginal properties of a signal, but rather about the asymptotic scaling of moments. Hence, although they provide useful information, they provide incomplete information — two quite distinct-looking signals could have an identical multifractal spectra.

**Multifractals From Multiplicative Cascades**

Cascades provide one of the simplest ways to create and understand multifractal measures [31–33]. The simplest cascade, the dyadic binomial measure, provides a good example of how cascades are constructed.

As illustrated in Figure 2.6, the process for constructing a dyadic binomial cascade involves an iterative product of random multipliers. We will denote these multipliers $M_k^j$, with $j$ denoting the construction stage and $k = 0, 1, \ldots, 2^j - 1$ denoting spatial location. We start with the uniform measure over the region $[0, 1)$ and multiply in a random variable $M_0^0$. To begin, we have

\begin{equation}
\text{Initialization: } \quad C(t) = M_0^0, \quad t \in [0, 1). \tag{2.42}
\end{equation}

At the first step, we split the measure into two regions $[0, 1/2)$ and $[1/2, 1)$. Over the first region, we multiply in a random variable $M_0^1$, and over the second region we multiply in a separate random variable $M_1^1$. Thus, at this step we have

\begin{equation}
\text{Stage 1: } \quad C(t) = \begin{cases} 
M_0^1 \cdot M_0^0, & t \in [0, 1/2), \\
M_1^1 \cdot M_0^0, & t \in [1/2, 1). 
\end{cases} \tag{2.43}
\end{equation}

Proceeding iteratively, we obtain after $L$ steps a distribution that is uniform on intervals
Figure 2.6: Iterative construction of the binomial cascade. After L stages of the construction, the measure \( C(t) \) is piecewise constant over the intervals \( [k2^{-L}, (k+1)2^{-L}) \), \( k = 0, 1, \ldots, 2^L - 1 \). The value of \( C(t) \) on each interval \( [k2^{-L}, (k+1)2^{-L}) \) is formed as product of \( L + 1 \) random multipliers \( M_k^L \).

\([k2^{-L}, (k + 1)2^{-L})\) and assigns to these intervals the mass

Stage \( n \): \( C(t) = M_{k_L}^L \cdot M_{k_{L-1}}^{L-1} \cdots M_{k_1}^1 \cdot M_0^0 \), \( t \in [k_L2^{-L}, (k_L+1)2^{-L}) \),

where the \( k_i, i = 1, \ldots, L - 1 \) can be derived from the choice of index \( k_L \). We can also write 2.44 in terms of the discrete-time signal \( C_{k_L}^{(L)} \) that we use to approximate \( C(t) \):

\[
C_{k_L}^{(L)} = \prod_{i=0}^{L} M_{k_i}^i.
\] (2.44)

Under fairly general conditions [89], this cascade construction leads to a process with multifractal properties as \( L \to \infty \). Note that the binomial of Figure 2.5(a) corresponds to (2.44) with \( M_i^L = \begin{cases} 1/4, & \text{i even} \\ 3/4, & \text{i odd} \end{cases} \).

A cascade is said to be conservative if the multipliers at each stage \( i > 0 \) satisfy the constraint \( M_{2n}^i + M_{2n+1}^i = 1 \), \( n = 0, 1, \ldots, 2^j - 2 \). For a conservative cascade, the total mass of the measure, \( M_0^0 \), is preserved at each step of the construction. We will show
in Chapter 5 that the Multifractal Wavelet Model construction corresponds to a binomial cascade.
Chapter 3

Wavelet-Domain Hidden Markov Models For Statistical Signal Processing

Our first primary goal is to accurately model the behavior of the wavelet coefficients of a wide class of signals. To this end, we adopt a statistical approach to wavelet-based signal modeling in which we regard the signal and its wavelet coefficients as random realizations from a family or distribution of signals. Thus, our objective is to develop wavelet-domain probability models that are rich and flexible enough to capture the structure of a wide variety of data, yet concise, tractable, and efficient for practical application in real-world problems. Models in hand, we can then perform a variety of statistical DSP tasks, such as estimation, compression, classification, and segmentation.

To construct these wavelet-domain probability models, we combine mixture models to capture the non-Gaussian property of wavelet coefficients with probabilistic graph theory to model dependencies between wavelet coefficients. Probabilistic graph theory, developed in the statistics and artificial intelligence community, allows us to specify independence relations between random variables via a graph model. In the graph model, we "connect" variables to establish dependency between them, obtaining conditional dependence relationships akin to those of Markov models. The power of graph theory is to allow us to model key dependencies, yet retain tractability and simplicity. Graph theory also tells us which models are self-consistent and which models can be easily and efficiently trained.

The combination of mixture models with probabilistic graph theory results in wavelet-domain hidden Markov models (HMMs), which are akin to the time-series Markov models used for speech recognition, but are adapted to the tree-structured topology of the wavelet transform. One model that we will focus on is the Hidden Markov Tree (HMT) model. We
Figure 3.1: Organization of a wavelet transform as a forest of binary trees. Tilings of the time-frequency plane and tree structures for (a) full decomposition (one tree), (b) decomposition with two fewer scale bands (four trees). We assign each wavelet coefficient in the tree an index \( i \). A scaling coefficient sits above the root of each tree.

develop novel Expectation Maximization (EM) algorithms to train and apply these models, with results that can be extended to other transforms with different types of time-frequency tilings. We then demonstrate the effectiveness of wavelet-domain HMMs for signal estimation (denoising) and classification. In particular, for signal estimation, we show that exploiting these key wavelet-domain dependencies leads to significant performance gains over standard independence assumptions [20].

3.1 Wavelet Trees

When viewed in the time-frequency plane as in Figure 2.1, a wavelet transform has a natural organization as a forest of binary trees [92]. The tree(s) are rooted at the wavelet coefficients in the coarsest scale (lowest frequency) band; a single scaling coefficient sits above each root. Depending on the length of the signal and the number of scale bands computed in the transform, the forest of trees will contain from one to several distinct trees (see Figure 3.1 and Figure 3.5). For instance, if we analyze a length-\( N \) discrete signal over \( L \) wavelet scales, we obtain \( N 2^{-L} \) wavelet trees.

To keep the notation manageable and consistent with the theory of probabilistic graphs (see Section 2.3), in this chapter we will adopt a graph-based index system for wavelet
atoms and coefficients: $\psi_{j,k} \rightarrow \psi_i$, $w_{j,k} \rightarrow w_i$. Thus a wavelet coefficient at scale $j$ and shift $k$ is now labelled with an index $i$. An example of this abstract indexing is shown in Figure 3.1(a) for the case of a single wavelet tree and in Figure 3.1(b) in the case of multiple trees. In our abstract indexing scheme, we will denote the $i$-th wavelet coefficient from the $k$-th tree as $w_i^k$.\footnote{We apologize in advance for the double-duty played by the index $k$ — as a shift $k$ in the standard notation and as the tree label $k$ in abstract notation, but since these notations are never combined we do not foresee any confusion from the two roles.} We will also use the notation $j = J(i)$ for the map that takes the index $i$ back to its corresponding scale $j$.

## 3.2 Residual Properties of Wavelet Coefficients

Having been thoroughly convinced from Section 2.2.2 that we should work in the wavelet domain, we must bear down and formulate some useful wavelet-domain models. Let us first review some previous work in this area.

Until recently, wavelet coefficients have been modeled either as jointly Gaussian [7, 14, 53, 57], or as nonGaussian but independent [1, 12, 82, 96]. Jointly Gaussian models can efficiently capture linear correlations between wavelet coefficients. However, Gaussian models are in conflict with the Compression property (Section 2.2.2), which implies the wavelet transforms of most signals are sparse, resulting in a large number of small coefficients and a small number of large coefficients. A typical wavelet coefficient density or histogram is thus much more “peaky” at zero and heavy-tailed than the Gaussian.

NonGaussian models have also been formulated, but usually the coefficients are assumed to be statistically independent of each other. Justification for independent non-Gaussian models is based on the primary properties plus the interpretation of the wavelet transform as a “decorrelator” that attempts to make each wavelet coefficient statistically independent of all others. However, the wavelet transform cannot completely decorrelate real-world signals — a residual dependency structure remains between the wavelet coefficients. In words, we have the following secondary properties of the wavelet transform:
Figure 3.2: Clustering and Persistence illustrated, respectively, in Donoho and Johnstone's (a) Doppler and (b) Bumps test signals [29]. The signals lie atop the time-frequency tiling (Figure 1.1 and Figure 2.1) provided by a seven-scale wavelet transform. Each tile is colored as a monotonic function of the wavelet coefficient energy $\|w_i\|^2$, with darker tiles indicating greater energy.

Clustering: If a particular wavelet coefficient is large/small, then adjacent coefficients are very likely to also be large/small [75].

Persistence: Large/small values of wavelet coefficients tend to propagate across scales [60, 61].

As we see in Figure 3.2, these are striking features of the wavelet transform. They have been exploited with great success by the compression community [75, 95]. The goal of this chapter is to do the same for signal processing.

3.3 Modeling Framework

Completely modeling the joint probability density function of all of the wavelet coefficients, $f(w)$ with $w = \{w_i\}$, would characterize the dependencies between the coefficients. However, the complete joint probability density is usually intractable to use and impossible to estimate. At the other extreme, modeling the wavelet coefficients as statistically independent, with $f(w) = \prod_i f(w_i)$, is simple but disregards the inter-coefficient dependencies. To strike a balance between these two extremes, we must represent the key
dependencies, and only the key dependencies. The primary and secondary properties of the wavelet transform suggest natural candidates: Persistence suggests that wavelet coefficients can have strong dependencies across scale (vertically in Figure 2.1), while Clustering and Locality suggest that coefficients can have strong dependencies within scale (horizontally in Figure 2.1).

We introduce a new modeling framework that neatly summarizes the probabilistic structure of the coefficients of the wavelet transform [19]. Our models owe their richness and flexibility to the following features:

**Mixture Densities:** To match the non-Gaussian nature of the wavelet coefficients, we model the marginal probability $f(w_i)$ of each coefficient as a mixture density with a hidden state variable.

**Probabilistic Graphs:** To characterize the key dependencies between the wavelet coefficients, we introduce Markovian dependencies between the hidden state variables. These dependencies are described by a probabilistic graph or tree (see Section 2.3).

Models of this type, commonly referred to as *Hidden Markov Models* (HMMs), have proved tremendously useful in a variety of applications, including speech recognition [22, 84] and artificial intelligence [97].

We turn to wavelet transform modeling using HMMs in Section 3.4. We discuss the training of these models in Section 3.5. In Section 3.6, we apply this machinery to several problems in signal estimation and detection and classification. We close in Section 3.8 with a discussion and conclusions.

### 3.4 Wavelet-Domain Probability Models

#### 3.4.1 Probabilistic models for a single wavelet coefficient

The Compression property of the wavelet transform states that the transform of a typical signal consists of a small number of large coefficients and a large number of small coefficients. More precisely, most wavelet coefficients have small values and hence contain
Figure 3.3: A two-state, zero-mean Gaussian mixture model for a random variable $W$. We denote the state variable $S$ with a white dot, the random variable $W$ with a closed dot. Illustrated are the Gaussian conditional pdf's for $W | S$ as well as the overall mixture pdf for $W$. In our application, we model each wavelet coefficient $W_i$ (each black dot in Figure 2.1) in this way.

very little signal information. A few wavelet coefficients have large values that represent significant signal information. This property leads to the following simple model for an individual wavelet coefficient. We model each coefficient as being in one of two states: "high," corresponding to a wavelet component containing significant contributions of signal energy, or "low," representing coefficients with little signal energy. If we associate with each state a probability density — say a high-variance, zero-mean density for the "high" state and a low-variance, zero-mean density for the "low" state — the result is a two-state mixture model for each wavelet coefficient.

As we see from Figure 3.3, the two-state, zero-mean mixture model is completely parameterized by the pmf of the state variable $S$, $p_S(1), 1 - p_S(1)$, and the variances of the Gaussian pdfs corresponding to each state. In most applications of mixture models, the value of the coefficient $W$ is observed, but the value of the state variable $S$ is not; we say that the value of $S$ is hidden.

Several factors support the model's validity. Empirical results from estimation have shown the this mixture model to be simple, yet effective [12, 82]. Our experience corroborates these results; in Figure 3.4 we demonstrate the fit that this model provides for an actual signal. Furthermore, theoretical connections have been made between wavelet coefficient mixture models and the fundamental parameters of Besov spaces — function spaces that have proved extremely useful for characterizing real-world signals [1].
Figure 3.4: A two-state, zero-mean Gaussian mixture model can closely fit real wavelet coefficient data. Here we compare the model pdf to the histogram of one scale of the wavelet transform of an image of fruit.

For any given set of wavelet data, the two-state, zero-mean Gaussian mixture model may not provide a fit to $f_w(w)$ with the desired fidelity. To improve accuracy, we can use Gaussian mixture models with $M > 2$ states or non-zero means in the Gaussian mixing densities. By increasing the number of states and allowing non-zero means, we can make the fit arbitrarily close for densities with a finite number of discontinuities [98]. We can even mix non-Gaussian densities, such as conditional densities belonging to the exponential family of distributions [85]. However, the two-state, zero-mean Gaussian mixture model is simple, robust, and easy-to-use — attractive features for many applications. For purposes of instruction, we will focus on the simple two-state model in this work, but develop machinery capable of handling more general mixture models.

In general, an $M$-state Gaussian mixture model for a random variable $W_i$ consists of

1. a discrete random state variable $S_i$ taking the values $m \in 1, 2, \ldots, M$ according to the pmf $p_{S_i}(m)$.

2. the Gaussian conditional pdfs $f_{W_i|S_i}(w|S_i = m)$, $m \in 1, 2, \ldots, M$.

To generate a realization of $W_i$ using the mixture model, we first draw a state value $m$ according to $p_{S_i}(m)$ and then draw an observation $w_i$ according to $f_{W_i|S_i}(w|S_i = m)$. The pdf of $W_i$ is given by

$$f_{W_i}(w) = \sum_{m=1}^{M} p_{S_i}(m) f_{W_i|S_i}(w|S_i = m).$$ (3.1)
Although each wavelet coefficient $W_i$ is conditionally Gaussian given its state variable $S_i$, the wavelet coefficient has an overall non-Gaussian density due to the randomness of $S_i$.

Unlike wavelet coefficients, scaling coefficients typically are not zero mean. Therefore, a two-state, zero-mean Gaussian mixture model may be inappropriate. One approach is to model the scaling coefficients as Gaussian with non-zero mean. Since scaling coefficients are essentially weighted averages of a large number signal samples, this approximation is reasonable in light of the Central Limit Theorem. A more flexible approach is to apply a Gaussian mixture model as before, but with non-zero-mean mixing densities.

3.4.2 Probabilistic models for a wavelet transform

Since a Gaussian mixture model can accurately characterize the pdf of a single wavelet coefficient, it seems logical to use Gaussian mixture models to characterize the joint pdf of the entire wavelet transform. The simplest approach would be to model the wavelet coefficients as independent Gaussian mixtures. We call this approach the *Independent Mixture (IM) model* (see Figure 3.5). Because the wavelet transform nearly decorrelates a wide variety of signals, this model for the wavelet tree is intuitively plausible. Moreover, as demonstrated by the denoising results in [12, 82], the IM model is a substantial improvement over deterministic signal models that do not explicitly take the distribution of signal’s wavelet coefficient values into account.

Nevertheless, the Clustering and Persistence properties lead to local dependencies between wavelet coefficients. Characterization of these dependencies has resulted in significant performance gains in compression [75, 95]. Ideally, we would like a model that both matches each individual coefficient’s pdf and captures dependencies between coefficients.

We motivate our approach by extending the Gaussian mixture model for one wavelet coefficient to jointly model two wavelet coefficients that represent components of the signal close in time and/or scale. We say that two such coefficients are neighbors. By Clustering and Persistence, if one coefficient is in a high-variance (low-variance) state, then its neighbor is very likely to also be in a high-variance (low-variance) state. Thus, the two neighbor-
ing wavelet coefficients can be modeled as Gaussian mixtures with *inter-dependent state variables*. This two-coefficient example suggests a natural generalization to the multiple coefficients in a wavelet transform: model each coefficient as a Gaussian mixture, but allow probabilistic dependencies between the state variables of each mixture.

What remains is to specify an appropriate model for these dependencies between the state variables. A complete joint pdf taking into account all possible dependencies is clearly intractable, since the number of different state variable combinations grows exponentially in the number of wavelet coefficients. Fortunately, the Locality and Multiresolution properties of the wavelet transform suggest that dependencies die off quickly as we move away from the local neighborhood about a coefficient of interest. Hence, very accurate and practical models can be obtained with probabilistic links between the states of only neighboring wavelet coefficients. We will now apply probabilistic graph theory (see Section 2.3) to develop these models.

**3.4.3 Graph models for wavelet transforms**

The Locality and Multiresolution properties of the wavelet transform suggest three simple ways to "connect the dots" representing the wavelet coefficients and states in Figure 2.1: (1) a graph with no dependencies between wavelet state variables, (2) a graph linking wavelet state variables across time using chains, and (3) a graph linking wavelet state variables across scale using trees. In Figure 3.5, we illustrate these graphs.

We are by no means limited to just these three graphs. We can develop graphs that capture even more inter-dependencies by placing additional connections between the states. Unfortunately, the computational complexity increases substantially for graphs more complicated than trees. Although we can still formulate algorithms for training and applying more general graphs [56,97], to keep our presentation and analysis simple we will concentrate on the three special cases described in Figure 3.5:

**Independent Mixture (IM) Model:** A mixture model with no connections, as in Figure 3.5(a), corresponds to the IM presented in [12,82] and discussed above. It treats
wavelet state variables (and hence wavelet coefficients) as independent random variables.

**Hidden Markov Chain Model:**

Connecting the state variables $S_i$ horizontally in Figure 3.5(b) specifies a Markov-1 chain dependency between the state variables within each scale [84]. This model treats wavelet state variables as dependent within each scale, but independent from scale to scale.

**Hidden Markov Tree (HMT) Model:** By connecting state variables vertically across scale in Figure 3.5(b), we obtain a graph with tree-structured dependencies between state variables. We call this new model a "tree model" to emphasize the underlying dependencies between parent and child state variables.

We will concentrate on the IM and HMT models in the sequel.

### 3.4.4 HMT model

The HMT model matches both the Clustering and Persistence properties of the wavelet transform. Its structure is reminiscent of the zerotree wavelet compression system [95],

\[ \text{Figure 3.5: Statistical models for the wavelet transform. (a) Independent Mixture (IM) model. To match the non-Gaussian nature of the wavelet coefficients, we model each coefficient as a mixture with a hidden state variable. Each black node represents a continuous wavelet coefficient } W_i. \text{ Each white node represents the mixture state variable } S_i \text{ for } W_i. \text{ (b) To match the inter-coefficient dependencies, we link the hidden states. Connecting discrete nodes horizontally across time (dashed links) yields the Hidden Markov Chain model. Connecting discrete nodes vertically across scale (solid links) yields the Hidden Markov Tree (HMT) model.} \]
which exploits tree-structured dependencies for substantial compression gains. Furthermore, this graph has a natural parent-child dependency interpretation. State variable dependencies are modeled via state transition probabilities from each parent state variable $S_i$ to its children’s states, the two state variables connected to it from below (if they exist). For example, in Figure 3.1(a), state variables $S_4$ and $S_5$ are both children of $S_2$, and hence causally dependent on $S_2$. Dependency is not simply limited to parent-child interactions, however. State variables $S_4$ and $S_5$ may be highly dependent due to their joint interaction with $S_2$.

Moreover, if we do not restrict ourselves to zero-mean, two-state Gaussian mixture model, but rather use Gaussian mixture models with more than two states and non-zero means, this simple tree-structure is capable of approximating the joint parent-child wavelet coefficient pdf to arbitrary precision [20].

Using an $M$-state Gaussian mixture model for each wavelet coefficient $W_i$, the parameters for the HMT model are:

1. $p_{S_1}(m)$, the pmf for the root node $S_1$.

2. $e_{i,p(i)}^{mr} = p_{S_i|S_{p(i)}}(m|S_{p(i)} = r)$, the conditional probability that $S_i$ is in state $m$ given $S_{p(i)}$ is in state $r$.

3. $\mu_{i,m}$ and $\sigma_{i,m}^2$, the mean and variance, respectively, of the wavelet coefficient $W_i$ given $S_i$ is in state $m$.

These parameters can be grouped into a model parameter vector $\theta$. Recall that we will primarily focus on the case $M = 2$ with the means $\mu_{i,m} = 0$. For a length-$N$ signal, this model translates to roughly $4N$ parameters, a number that can be greatly reduced by “tying,” as we will see in Section 3.5.3.

In the HMT model, we have the following conditional independence relationships among the wavelet coefficients $\{W_i\}$. First, we observe that

$$f_{W_i}(w_i|\{W_j\}_{j \neq i}, \{S_j = s_j\}_{j \neq i}, S_i = s_i) = f_{W_i}(w_i|S_i = s_i).$$

(3.2)
In words, \( W_i \) is conditionally independent of all other random variables given its state \( S_i \). Hence, the conditional independence properties for the states also lead to conditional independence properties for the wavelet coefficients.

It is important to note that the Markov structure is on the states of the wavelet coefficients, not on the coefficients themselves. This is an important distinction between our model and other multiscale Markov signal representations such as those considered in [7, 57]. Because the states are never known exactly, our HMM framework does not place a Markov structure on the wavelet coefficients directly. However, even though the wavelet coefficients are generally not Markov, signal processing using wavelet-domain HMMs remains efficient due to the Markov nature of the wavelet state variables.

3.4.5 Extension to multi-dimensional data

Although for illustrative purposes we have focused on one-dimensional signals, our results apply to wavelet-domain HMMs for images and multi-dimensional data as well. For example, HMT models for one-dimensional signals have a natural binary tree structure, with each wavelet state connected to the two "child" wavelet states below it (see Figure 3.5(b)). HMT models for images have a natural quadtree structure, with each wavelet state connected to the four "child" wavelet states below it (see Figure 3.6). HMT models for \( m \)-dimensional data have a natural \( 2^m \) tree structure. See [13, 90] for related work in this area.

3.4.6 Three standard problems of HMMs

There are three canonical problems associated with the wavelet-domain HMMs we have described [84]:

**Training:** Given one or more sets of observed wavelet coefficients \( \{ w_i \} \), determine the wavelet-domain HMM parameters \( \theta \) that best characterize the wavelet coefficients.

**Likelihood Determination:** Given a fixed wavelet-domain HMM with parameters \( \theta \), de-
Figure 3.6: HMT for an image quadtree. As shown for two parent states, each parent hidden state is connected to its four child states. (The other parent-child connections are omitted for visual clarity. The two remaining parent states would have similar connections to their child states.) The two fields of nodes depict the wavelet coefficients at scales \( j \) and \( j + 1 \), respectively, and correspond to 2-d wavelet basis functions with a specific spatial orientation (horizontal, vertical, or diagonal). For more details on 2-d wavelet systems see [21, 95, 107].

Determine the likelihood of an observed set of wavelet coefficients \( \{w_i\} \).

**State Estimation:** Given a fixed wavelet-domain HMM with parameters \( \theta \), determine the most likely sequence of hidden states \( \{s_i\} \) for an observed set of wavelet coefficients \( \{w_i\} \). This is useful for problems such as segmentation (see [9]), where the hidden states represent a physically meaningful quantity.

We next focus on training and likelihood determination, since they are crucial for the applications that we develop in Section 3.6.

### 3.5 Training and Likelihood via the EM Algorithm

In training, we seek the parameters of a wavelet-based HMM that best fit a given set of data. The training data \( w = \{w_i\} \) consists of the wavelet coefficients of an observed signal(s); the model parameters \( \theta \) consist of the mixture state probabilities and the mean and variance of each Gaussian component. For parameter estimation, we apply the maximum likelihood (ML) principle. ML estimates are asymptotically efficient, unbiased, and consistent as the number of training observations increases.
Direct ML estimation of model parameters $\theta$ from the observed data $w$ is intractable, since in estimating $\theta$ we are characterizing the states $S = \{S_i\}$ of the wavelet coefficients $w$, which are unobserved (hidden).² Yet, given the values of the states, ML estimation of $\theta$ is simple (merely ML estimation of Gaussian means and variances). Therefore, we employ an iterative *Expectation Maximization* (EM) approach [24], which jointly estimates both the model parameters $\theta$ and probabilities for the hidden states $S$ given the observed wavelet coefficients $w$. In the context of HMMs, the EM algorithm is also known as the *Baum-Welch* algorithm.

### 3.5.1 EM algorithms for training

Our discussion of EM algorithms focuses on the specific problem of parameter estimation in wavelet-based HMMs; for a more general treatment, see [24]. We begin with some terminology. The *incomplete* data is our training data $w$, and the *complete* data $(w, s)$ is our training data augmented with the hidden states $s$. Our goal is to maximize the incomplete log-likelihood function $\ln f(w|\theta)$. The EM algorithm decouples this difficult maximization into iteration between two simpler steps: the E step and the M step.

At the $l$th iteration, the E step calculates the expected value $E_S \left[ \ln f(w, S|\theta) \mid w, \theta^l \right]$. The M step then maximizes this expression as a function of $\theta$ to obtain $\theta^{l+1}$ for the next iteration. Under mild conditions, this iteration converges to a local maximum of the log-likelihood function $\ln f(w|\theta)$ [24]. Efficient EM algorithms for HMMs exist under the assumption that the underlying probabilistic graph is chordal [56, 97]. A graph is chordal if all cycles of length greater than 3 have a chord.³ Since the HMMs considered in this work do not contain cycles, they are trivially chordal and hence admit efficient EM training.

For HMMs, the complexity of one EM iteration is linear in the number of observa-

²Since the states are not observed, we will generally denote them using capital letters to emphasize their uncertain or random nature.

³Recall from Section 2.3 that a *cycle* of a graph is a path starting and ending at the same node — a closed-loop. A *chord* is a connection between two non-consecutive nodes in a cycle.
tions [56, 84]. With intelligent initialization, the algorithm can converge in as few as ten iterations for a simple two-state HMT model. However, as the graph model underlying an HMM becomes more complicated, each iteration of the EM algorithm becomes more computationally intensive (still linear complexity, but with a large constant factor), and the algorithm may take longer to converge. Hence, it is important to keep the HMM as simple as possible.

The specific EM steps for the IM and hidden Markov chain models have been developed thoroughly in [84, 85], so we do not include them here. For more general tree models, Ronen et al. provide specific EM steps for discrete variables in [91]. Since the observed wavelet data in the HMT model is continuous-valued, we provide a new EM algorithm for this model in Appendix A.

3.5.2 Likelihood determination

The E step of the EM algorithm is useful in its own right, since it calculates \( \ln f(w|\theta) \), the log-likelihood of the observed data given the model. This likelihood calculation basically measures how well the model \( \theta \) describes the data \( w \). Hence, it is useful for detection and classification applications, as we will see in Section 3.6.3. The calculation can also be used to predict or estimate the values of wavelet coefficients given the model.

3.5.3 Robust training via tying

HMMs are very rich models; thus we must ensure that we have enough training data to prevent "overfitting." By averaging over only one or very few signal observations, we cannot expect to robustly estimate the marginal densities of the wavelet coefficients, let alone a joint density for the entire wavelet transform. This brings us to a key problem: If limited training data are available, how can we make our modeling more robust? We do so by modeling random variables that have similar properties using a common density or a common set of density parameters. For example, if we expect two random variables to have roughly the same variability, we can describe them with a common variance. In this way, we ob-
taining more reliable parameter estimates by increasing the amount of training data associated with each parameter. This practice is known as “tying” in the HMM literature [84], and we use it to more robustly estimate the means, variances, and transition probabilities of our wavelet-domain HMMs.

In Figure 3.7, we distinguish between two different types of tying in the HMT model, tying between wavelet trees and tying within wavelet trees. Recall from Section 3.1 (see also Figure 3.5) and Figure 3.1) that in general, the wavelet decomposition of even a single signal observation can result in multiple wavelet trees. By tying across trees — which assumes that the coefficients of these trees have the same density — we can train as if we had multiple signal observations. We can also tie within trees — by tying all coefficients within the same scale of a tree, for example. Tying within scale reduces the number of parameters from $4N$ to $4L$, with $N$ the data length and $L$ the number of the scales of the wavelet transform.

### 3.6 Applications

Our development of wavelet-domain HMMs has been motivated by the intrinsic properties of the wavelet transform, and we have discussed how several aspects of the model are supported by empirical and theoretical evidence. However, the true test of our modeling framework lies in its application to real signal processing “benchmark” problems. To this
Figure 3.8: Block diagrams for wavelet-based denoising. (a) Standard wavelet-based denoising. (b) Our empirical Bayesian wavelet-based denoising. In each case, the signal in additive white Gaussian noise (WGN) is wavelet-transformed, passed through threshold-like nonlinearities, and inverse transformed to get the denoised signal. However, the empirical Bayesian approach attempts to learn and exploit the signal’s wavelet-domain structure to better separate signal from noise.

end, we consider applications in signal estimation and detection/classification.

We compare the estimation performance of our new models for signal estimation in additive noise to state-of-the-art wavelet denoising methods. We show that our new framework offers significant improvements in several well-studied benchmark problems. Wavelet-domain HMMs are also well-suited to signal detection and classification.

In this Section, we approach these problems by assuming that no prior signal models are available and that only “training” data are available for the design of the detector/classifier. We compare the wavelet-domain HMM-based detectors to classical detectors. Our results demonstrate the HMM’s high performance and extremely efficient use of training data in two difficult signal detection problems.

3.6.1 Wavelet-based estimation framework

Wavelets have proved remarkably successful for estimating signals in additive white Gaussian noise (WGN) [12, 29]. The Compression property indicates that the wavelet transform typically compacts signals into just a few coefficients of large magnitude. Because the wavelet transform is orthogonal, it leaves white noise evenly distributed across many coefficients of small magnitude. Therefore, by setting small wavelet coefficients to zero, one effectively removes noise without degrading the signal (see Figure 3.8(a)).
Existing denoising methods usually ignore possible dependencies between signal wavelet coefficients, and hence these methods do not exploit key Clustering and Persistence properties. In this Section, we illustrate the power of the HMT model by developing a novel signal denoising method based on this framework. The new denoising method coordinates the noise removal among the wavelet coefficients and automatically adjusts to subtle structure within the signal [19] (see Figure 3.8(b)).

Consider the problem of estimating a length-$N$ signal $x$ in zero-mean white Gaussian noise with power $\sigma_n^2$. Taking the $L$-scale wavelet transform of the noisy signal, we obtain $K = N 2^{-L}$ trees of noisy wavelet coefficients $\{w_i^k\}$ (see Section 3.1). Since the orthogonal wavelet transform of zero-mean white Gaussian noise is zero-mean white Gaussian noise of the same power, the estimation problem can be expressed in the wavelet domain as

$$w_i^k = y_i^k + n_i^k,$$

(3.3)

where $w_i^k$, $y_i^k$, and $n_i^k$ denote the wavelet coefficients of the observed data, the signal, and the noise, respectively.

### 3.6.2 Estimation using wavelet-domain HMMs

Our approach is succinctly described as follows. We first fit a HMT model to the $y_i^k$'s from the noisy data and then use this model as a prior signal distribution to compute the conditional mean estimates of the $y_i^k$'s given $w_i^k$. In effect, this approach is an "empirical" Bayesian estimation procedure under squared-error loss. It is empirical, since we estimate the parameters of our Bayesian prior from the data itself (see Figure 3.8(b)). To fit an HMT to the noisy wavelet coefficients, we apply the EM algorithm from the Appendix. We begin by estimating the parameters $\{p_{S_1}(m), \ e_{i,\rho(i)}^{m}, \sigma_{i,m}^2\}$ for the signal wavelet coefficients using the noisy signal observation.\(^4\)

\(^4\)As in [12, 82], we assume that the wavelet coefficients are zero-mean. The scaling coefficients, though not zero-mean, are relatively noise-free and hence are not processed.
The key observation is that if the signal has a wavelet-domain HMM pdf, then the noisy signal does as well. This observation stems from two facts. First, the sum of independent Gaussian random variables is also Gaussian, with variance the sum of the two variances. Second, given the values of their hidden state variables, the signal wavelet coefficients are Gaussian. Therefore, adding the independent zero-mean white Gaussian noise $n_i^k$ increases each mixture model variance $\sigma_{i,m}^2$ by $\sigma_n^2$, but leaves the other parameters unchanged. Hence, we can obtain the signal wavelet model from the noisy signal by fitting an HMM to the noisy signal wavelet coefficients and then subtracting the added variance due to noise. If we denote the mixture variance of the noisy wavelet coefficient at location $i$ in the $m$th state as $\gamma_{i,m}^2$, then

$$\sigma_{i,m}^2 = \left(\gamma_{i,m}^2 - \sigma_n^2\right)_+, \quad (3.4)$$

with $(x)_+ = x$ for $x \geq 0$ and $(x)_+ = 0$ for $x < 0$. The noise power $\sigma_n^2$ can be estimated using the median estimate of [29] performed on the finest scale wavelet coefficients (where the signal energy is expected to be negligible).

Of course, we typically have only a single noisy signal observation at hand. Therefore, in order to insure reliable parameter estimation for the signal we must "share" statistical information between related wavelet coefficients. We accomplish this by assuming that all wavelet coefficients and state variables within a common scale are identically distributed, including identical parent-child state transition probabilities. (This model corresponds to tying both within and across trees from Section 3.5.3). The resulting HMT model is completely parameterized by two mixture variances for the wavelet coefficients at each scale, two probabilities for the root state variable at the coarsest scale, and $2 \times 2$ state transition probability matrices for the state variables at all other scales.

Once we have trained the HMT, estimation of the true signal wavelet coefficients (denoising) is straightforward. Note that if the states $S_i^k$ of the signal wavelet coefficients $y_i^k$ are known, then the estimation problem becomes a series of simple one-dimensional problems of estimating zero-mean Gaussian random variables in zero-mean additive Gaussian
noise. The conditional mean estimate of \(y_i^k\), given \(w_i^k\) and the state \(s_i^k\), is

\[
\mathbb{E}[Y_i^k | W_i^k = w_i^k, S_i^k = m] = \frac{\sigma^2_{i,m}}{\sigma_i^2 + \sigma_{i,m}^2} w_i^k.
\] (3.5)

Now recall that by-products of the EM algorithm are the hidden state probabilities \(p(S_i^k | w_i^k, \theta)\) given the model and the observed wavelet coefficients. (See the Appendix for how these probabilities are calculated.) Using these state probabilities, we obtain conditional mean estimates for \(y_i^k\) via the chain rule for conditional expectation

\[
\mathbb{E}[y_i^k | w_i^k, \theta] = \sum_m p(S_i^k = m | w_i^k, \theta) \times \frac{\sigma^2_{i,m}}{\sigma_i^2 + \sigma_{i,m}^2} w_i^k.
\] (3.6)

The final signal estimate (denoised signal) is computed as the inverse wavelet transform of these estimates of the signal wavelet coefficients. Note that only the wavelet coefficients are processed. The original scaling coefficients are used in the inverse transform.

We now compare our "empirical" Bayesian denoising procedure using the IM and HMT with current state-of-the-art wavelet denoising algorithms.\(^5\) Table 3.1 compares the estimation performance of the IM and the HMT models with two state-of-the-art scalar algorithms. Donoho and Johnstone's SureShrink algorithm [29] performs scalar thresholding in the wavelet domain. The Bayesian mixture algorithm of Chipman et al. [12] operates in a similar fashion to the denoising method using the IM model, except that their mixture model is a true Bayesian prior and is not inferred from the data. Mean-squared-error (MSE) results are tabulated for denoising Donoho and Johnstone's length-1024 test signals Bumps, Blocks, Doppler, and Heavisine [29] in additive white Gaussian noise of power

\(^5\)For each estimation algorithm, Bumps was transformed using the Daubechies-4 wavelet. Blocks using the Haar wavelet, and Doppler and Heavisine using the Daubechies-8 most-nearly-symmetric wavelet. The IM and HMT algorithms used a seven-scale wavelet decomposition. The error results of Table 3.1 for SureShrink and the Bayesian algorithm of Chipman et al. were quoted from [12]. More details of these two algorithms are provided in [12, 29]. Error results for IM and HMT were obtained by averaging over 1000 trials. For Figure 3.9, SureShrink was implemented using the "hybrid" shrinkage estimator in the WaveLab software. The Bayesian mixture algorithm [12] was not implemented for Figure 3.9, but is similar to IM both in its Bayesian formulation and MSE performance.
Figure 3.9: Denoising the Doppler test signal in white Gaussian noise, $\sigma_n^2 = 2.25$. On each plot a dotted line is used to depict the original signal, a solid line the noisy or denoised signal. The leftmost plots depict the entire signals; the rightmost plots depict the signals "zoomed" to the interval [0,0.04], where it is difficult to distinguish high-frequency signal from noise. (a) Noisy length-1024 Doppler signal, MSE = 2.42. (b) Denoised via SureShrink [29], MSE = 0.43. (c) Denoised via wavelet-based Bayesian IM model, MSE = 0.34 (d) Denoised via wavelet-based Bayesian HMT model, MSE = 0.26.
Table 3.1: Denoising results for Donoho and Johnstone's length-1024 test signals [29]. Noise variance \( \sigma_n^2 = 1 \). (The signals were scaled to have a time-domain variance of 49).

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean-squared error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bumps</td>
</tr>
<tr>
<td>SureShrink [29]</td>
<td>0.683</td>
</tr>
<tr>
<td>Bayesian [12]</td>
<td>0.350</td>
</tr>
<tr>
<td>IM</td>
<td>0.335</td>
</tr>
<tr>
<td>HMT</td>
<td>0.268</td>
</tr>
<tr>
<td>Context 1</td>
<td>0.252</td>
</tr>
<tr>
<td>Context 2</td>
<td><strong>0.249</strong></td>
</tr>
</tbody>
</table>

\( \sigma_n^2 = 1 \). Inspection of Table 3.1 shows that significant MSE gains can be achieved by exploiting wavelet-domain dependencies via the HMT model. (The algorithms and results for Context 1 and Context 2 will be discussed in Chapter 4.) The only exception is the Heavisine signal, which has less wavelet-domain structure for the HMT model to exploit. In this case, the IM and HMT model perform roughly equivalently.

Figure 3.9 illustrates the subjective improvement\(^6\) of the HMT model for denoising a signal realization of the Doppler signal in white Gaussian noise of power \( \sigma_n^2 = 2.25 \). We see that the HMT denoising method offers two significant advantages over the other methods: (1) HMT denoising is often smoother than both SureShrink and IM, and (2) HMT denoising preserves the high-frequency components at the beginning of the signal better than the other methods. This demonstrates how exploiting the statistical dependencies between wavelet coefficients enables HMT denoising to better separate signal from noise — even in regions where signal and noise are visually indistinguishable.

\(^6\)Results vary depending on the noise realization, and there can be no guarantee of smoothness with any MSE-based optimality criterion. However, the figure is a typical example of the subjective improvements that can result from exploiting dependencies between wavelet coefficients.
3.6.3 Wavelet-based detection and classification framework

Our marriage of wavelet transforms and HMMs yields a flexible framework for generalized likelihood-based signal detection and classification that both matches the properties of the wavelet transform and exploits the structure inherent in real-world signals. Given iid signal observations from two or more classes of signals, we can train HMMs for each class \( c \), resulting in parameter vectors \( \theta_m \). We use the trained HMMs to detect or classify a new signal observation \( w \) by determining which describes the new observation best. This task boils down to computing the likelihood of the new signal observation for each HMM \( f(w|\theta_m) \) and then selecting the class \( m \) whose HMM provides the greatest likelihood. This approach is analogous to the use of HMMs for speech recognition [22], where each signal class is a specific word or utterance. A slightly different approach developed for time-domain HMMs has been shown to be asymptotically optimal in the Neyman-Pearson sense for two-class problems [69].

Several other wavelet-based detection and classification schemes have been proposed [53, 74, 92, 108]. Our purpose is not to provide a comprehensive review of wavelet-based detection algorithms, but rather to demonstrate the potential of the new wavelet-domain HMM framework for signal detection and classification. Note, however, that this approach is quite different from the other wavelet-based detection schemes mentioned above. An related image segmentation algorithm using wavelet-domain HMMs was developed in [13].

The properties of the wavelet transform make our framework particularly appropriate for the classification and detection of real-world signals. To demonstrate the power and potential of wavelet-domain HMMs for signal classification, we tackle two difficult problems — classification of nonlinear processes and change detection. These problems arise in many applications, including sonar and radar, machinery and process monitoring, and biomedical signal analysis. We do not suggest that this framework is the optimal one for either specific problem, rather we chose these two examples to demonstrate the flexibility and adaptability of the approach. In situations where the data is known to obey a simple probability model, then optimal detection and classification methods should be used. However,
in complicated real-world applications where the only prior information is a set of training data, our approach offers a useful framework for detection and classification. In combination, wavelet HMMs and training data provide an efficient and powerful framework for generalized likelihood ratio testing. Both examples considered here are binary hypothesis problems, but the framework is applicable to multiple hypothesis testing as well.

In these examples, we applied a Haar transform with a single wavelet tree and a single scaling coefficient. We modeled the wavelet coefficients using two-component \((M = 2)\) IM and HMT models with non-zero mixture means. These models were trained using multiple signal observations (without tying). We did not model the scaling coefficient, since it provides the global mean of the signal, which in both examples was the same under the two hypotheses. In other scenarios the scaling coefficient(s) may provide vital discriminating information.

For the purposes of illustration, we only considered a very special form of the wavelet-domain HMM framework — one based on the Haar transform and two-component mixture models. Although the Haar transform is appropriate for edge detection, different wavelet transforms may work better for other applications. Generally, we expect the transform that best compresses the signals of interest to provide the best performance. In addition, we could use more flexible models with \(M > 2\) mixture components, but in using such models we would risk "overfitting" the training data.

### 3.6.4 Classification and detection of nonlinearity

For the purposes of demonstration, we have designed a numerical experiment that captures many of the nuances that make nonlinearity classification/detection so difficult. We consider two classes of random processes described mathematically by:

\[ I: \quad x_1[k] = ax_1[k - 1] + n_1[k] \]

\[ \Pi: \quad x_2[k] = y_2[k] + 0.2y_2^2[k], \]

with \(y_2[k] = by_2[k - 1] + n_2[k]\).
Both $n_1$ and $n_2$ are white Gaussian noise processes, and the autoregressive (AR) parameters $a$ and $b$ are iid and uniform over the interval $(0.4, 0.8)$. The signals are discrete-time and organized into signal vectors of length 128 with $(k = 1, 2, \ldots, 128)$. Class I signals are linear AR(1) processes. Class II signals are produced by passing linear AR(1) processes through a memoryless cubic nonlinearity. Examples of signals from each class are shown in Figure 3.10 (generated with the same AR parameter and white noise excitation for comparison).

The first task at hand is to train wavelet-domain HMMs for the two classes based on labeled observations from each class. We generated $N_T$ iid AR signals from each class for training purposes. (Note: the AR parameter was varied independently for each realization.) For comparison, we constructed a minimum-probability-of-error quadratic detector under the assumption that the two classes have Gaussian distributions with different means and covariances [93], with the means and covariances estimated from the training data. The quadratic detector is not optimal, since the second class is nonGaussian. In cases where the number of training observations $N_T$ was smaller than the dimension of the observations, we formed the quadratic detector in the subspace spanned by the training data.

After training the classifiers, we tested their performance with 1000 additional iid observations from each class. To obtain reliable estimates of the error rates, we repeated the
training and testing procedure 10 times in each case. The error rates for the IM model, HMT model, and quadratic detector, as a function of the number of training vectors $N_T$ from each class, are shown in Figure 3.11.

Given a limited amount of training data, the quadratic detector had a difficult time distinguishing the classes and thus offers very poor performance. In contrast, the wavelet-domain HMMs make much more efficient use of the training data. With only 128 training vectors from each class, the performances of the HMMs have stabilized to their minimum error rates. Additional training data does not increase their performance. The performance of the quadratic detector does improve as $N_T$ increases, but requires nearly 10 times the amount of training data that the HMMs require for the same error rate. We see that asymptotically (in number of training data $N_T$) the quadratic detector has the best error rate, followed closely by the HMT model. The IM model has the worst asymptotic error performance. This demonstrates the performance gains associated with the HMT model. Also, this suggests that more complex wavelet-domain HMMs (that is, more probabilistic connections between states) may provide asymptotic performances that meet or even exceed that of the quadratic detector. Of course, more complex HMMs will also require more training data to achieve such performance. These and related issues are currently under investigation.
Figure 3.12: Typical signals for the abrupt change detection experiment. (a) Gaussian white noise added to constant signal (Class I). (b) Gaussian white noise added to signal with abrupt change (Class II).

3.6.5 Detection of an abrupt change

In this example, we consider the following two-class problem. Class I consists of random discrete-time processes with an arbitrary mean value and additive white Gaussian noise. Class II consists of random discrete-time processes with an abrupt arbitrary change in the mean at some arbitrary point in the signal. Again, our signal observations are organized into length-128 observation vectors. Formally, our signal classes are defined by:

\[
\text{I:} \quad x_1[k] = a_1 + n_1[k] \\
\text{II:} \quad y_2[k] = a_2 I_{[1, \tau]}[k] + b_2 I_{[\tau+1, 128]}[k] + n_2[k].
\]

Both \(n_1\) and \(n_2\) are white Gaussian noise processes, and \(a_1, a_2,\) and \(b_2\) are iid and uniform on \([-1, 1]\). \(I_{[t_1, t_2]}[k] = 1\) if \(t_1 \leq k \leq t_2\) and is zero otherwise. The change-point \(\tau\) is uniformly distributed over the integers \(\{16, \ldots, 112\}\). Examples of signals from each class are shown in Figure 3.12.

An excellent treatment of classical methods for the detection of abrupt changes is given in [8]. In addition, other wavelet-based approaches to the change point problem have been discussed in the literature [74, 108]. The purpose of this example is not to make an exhaustive comparison between our method and other existing techniques in the literature; rather,
the intent is simply to demonstrate the versatility of the wavelet-based HMM approach to signal detection.

We again designed the wavelet-domain HMM classifiers (Haar-based IM model and Haar-based HMT model) with training data from each class, and then tested their performance with 1000 additional iid observations from each class. The error rates for the IM model and HMT model as a function of the number of training vectors $M$ from each class, are shown in Figure 3.13. For comparison, we also implemented an “off-line” generalized likelihood ratio test (GLRT) detector [8]. Although the signal parameters (position and size of change) and noise variance are estimated from the data itself, the GLRT detector makes full-use of piecewise constant signal model and Gaussian noise model. Since the GLRT exploits all prior information, we may view the performance of the GLRT detector as a lower bound on detection performance in this case. Note that with just a few hundred training samples, the performances of the wavelet-domain HMM detectors approaches that of the GLRT detector.

Clearly the HMMs are not the optimal detector in this problem. With precise knowledge of the problem at hand, more efficient detectors such as the GLRT are easily designed. However, this experiment again demonstrates the utility of the wavelet-domain HMMs for
Figure 3.14: A directed graph corresponding to an upwards HMT model. The direction on the edges move from the leaves to the root, implying the roles of parent and children are reversed from the standard HMT model in Figure 3.5(b).

modeling data with little or no prior information.

3.7 Persistence and An Upwards HMT

The HMT model is equivalent to a directed tree graph with directions that point from the root to the leaves. By reversing the directions from leaves to root, in Figure 3.14, we form an "upwards HMT" in which the roles of parent state and child state become reversed. Since the definition of parent and child depend on whether a tree is directed upward or downward, let us denote the wavelet coefficient (or state) at the coarser scale by "branch" and the two wavelet coefficients (or states) connected to it from below by "twigs." Recalling from Figure 2.3(b) the implications of pointing the edges from leaves to root, we see that an upwards HMT has the property that the twig states are independent a priori, but dependent given the branch state.

By inspecting Figure 3.2, we observe that the wavelet-domain energy due to an edge narrows as we move from coarser to finer scales. Thus, if a branch wavelet coefficient is large, we expect that one of its two twig coefficients is likely to be large — not both. An upwards HMT is better-equipped than the standard HMT to model this behavior. In the upwards HMT, if the branch state is large, we can model the fact that only one of the two
twig states should be large.

We tested the hypothesis that if a branch wavelet coefficient is large that one, and only one, of its twigs coefficients is likely to be large. We did so by calculating the probability that both the twig coefficients were large given a large branch coefficient under the following two assumptions: (a) dependency is allowed between twigs and (b) the twigs are assumed to be independent given the branch. Note that (b) corresponds to the assumption of the HMT model. The deviation of (b) from (a) corresponds to inaccuracy in the independence assumption. If the probability from (b) is greater than (a), then the independence assumption ignores an inverse relationship between the twigs. If the probability from (b) is less than (a), then the independence assumption ignores a positive dependency between the twigs.

We performed the test on Donoho's four test signals Blocks, Bumps, Doppler and Heavisine [29] using the same wavelet filters as the example of Table 3.1. For each test signal, at each wavelet scale we set a threshold to be the median of the energy at that scale. All wavelet coefficients with squared magnitude above the threshold were labelled "large," and all wavelet coefficients with squared magnitude smaller than the threshold were labelled "small."

For case (a), the probability of a transition from a large branch to two large twigs was calculated from the number of times that the branch and its two twigs were all large. For case (b), the probability of a transition from a large branch to a large twig was calculated from the number of times that the branch and one of its twigs were large. The probability of a transition from a large branch to two large twigs was found by squaring the result. The findings, presented in Table 3.2, were at first surprising.

In most cases, when we allow for dependence between the twig states, we observe that given the branch is large, the two twig states have a slightly greater probability of being jointly large. This is the opposite of what we would expect from Figure 3.2. The pictures "lie:" our eyes measure size relative to a fixed threshold across the entire transform, whereas the HMT model measures size relative to within-scale energy, which decays as we
Table 3.2: For Donoho’s length-1024 test signals [29], we depict the probabilities that two twigs are large given the branch is large under a dependence assumption (Dep.) and under an independence assumption (Indep.). Under the independence assumption, the twigs are assumed to be independent given the branch, as in the standard HMT independence assumption (HMT). The probabilities are calculated for each of wavelet scales 5-9, with 9 the finest scale.

<table>
<thead>
<tr>
<th>Wavelet scale</th>
<th>Doppler</th>
<th>Bumps</th>
<th>Heavisine</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.750</td>
<td>0.660</td>
<td>0.500</td>
<td>0.562</td>
</tr>
<tr>
<td>6</td>
<td>0.812</td>
<td>0.766</td>
<td>0.875</td>
<td>0.766</td>
</tr>
<tr>
<td>7</td>
<td>0.844</td>
<td>0.793</td>
<td>0.938</td>
<td>0.908</td>
</tr>
<tr>
<td>8</td>
<td>0.922</td>
<td>0.864</td>
<td>0.938</td>
<td>0.938</td>
</tr>
<tr>
<td>9</td>
<td>0.969</td>
<td>0.954</td>
<td>0.992</td>
<td>0.992</td>
</tr>
</tbody>
</table>

move to finer scales. For example, a fine-scale wavelet coefficient could be large compared to its fine-scale brethren, but on an absolute scale could appear small.

For the Blocks signal, the two twig states do have a lower probability of being jointly large. This is because the edges in the block signal are perfect step discontinuities that “line-up” with only one of the two Haar wavelet basis functions at each scale. In general cases — when the Haar is not used and the edges are not perfect step discontinuities — we do not expect this behavior to be prevalent.

From this simple test, we expect that an upwards HMT, while interesting, will not significantly improve modeling accuracy in comparison to the standard HMT. This conclusion has been verified by preliminary experiments.

### 3.8 Discussion

The primary properties of the wavelet transform — Locality, Multiresolution, and Compression — have led to powerful new approaches to statistical signal processing. However, existing methods usually model the wavelet coefficients as statistically independent or jointly Gaussian. The Compression property dictates the need for nonGaussian mod-
els for individual wavelet coefficients. Moreover, the secondary properties of the wavelet transform — Clustering and Persistence — indicate that statistical dependencies between coefficients must be characterized in order to derive optimal signal processing algorithms.

In this work, we have developed a new framework for statistical signal processing based on wavelet-domain HMMs. The framework enables us to concisely model the non-Gaussian statistics of individual wavelet coefficients and capture statistical dependencies between coefficients. We have developed an efficient Expectation Maximization algorithm for fitting the HMMs to observational signal data, and we have demonstrated the utility, flexibility, and performance of our framework in several estimation and detection problems.

The HMM framework presented here could serve as a powerful new tool for wavelet-based statistical signal and image processing, with applications in signal estimation, detection, classification, compression, and synthesis. Although the examples we have provided here are one-dimensional, multi-dimensional wavelet domain HMTs are easily derived from our results, since the models and training algorithms apply to quad and higher-dimensional trees. Furthermore, these HMMs apply not only for modeling wavelet-domain data, but also for modeling data from other multiresolution transforms or signal representations. Finally, the knowledge base that has already accumulated in statistics, speech recognition, artificial intelligence, and related fields may lead to wavelet-domain HMMs that are even more accurate and sophisticated, yet still tractable, robust, and efficient for signal processing.
Chapter 4

Context-Based Hidden Markov Models For Statistical Signal Processing

Although the HMT model is powerful and relatively simple, in certain applications it is crucial to model more and different dependencies between the wavelet coefficients. More sophisticated dependency structures for the hidden states can be formulated using the theory of probabilistics graphs, but the analysis and training of more complicated HMMs becomes much more complicated and computationally intensive. One reason is that these graphs models lack a single causal direction. A single causal direction (such as forward through time in a Markov chain and down the tree in an HMT) allows dependencies to be expressed simply and effectively through the use of state transition probabilities. However, by modeling wavelet coefficient inter-dependencies via contexts, we retain the approximation capabilities of HMMs, yet substantially reduce their complexity. To illustrate the power of this approach, we develop new algorithms for signal estimation and for efficient synthesis of non-Gaussian, long-range-dependent network traffic.

4.1 Definition and Use of Contexts

We define the context for $W_i$ as a length-$P$ vector $V_i \equiv [V_{i,1}, V_{i,2}, \ldots, V_{i,P}]$ formed as a function of the wavelet or scaling coefficients (see Figure 4.1). We condition $S_i$ on $V_i$ to predict $W_i$. The idea is for $V_i$ to provide supplementary information to the HMM, so that given the context, we can treat the wavelet coefficients as independent.

By conditioning (3.1) on $V_i$ (with the added assumption that $V_i$ and $W_i$ are independent
given $S_i$, we have the context-based mixture model for the wavelet coefficients:

$$ f_{W_i|V_i}(w|v_i) = \sum_{m=1}^{M} p_{S_i|V_i}(m|v_i) f_{W_i|S_i}(w|S_i = m). \quad (4.1) $$

In this case, the mixing probabilities depend on the value of the context $V_i$. If $V_i$ is highly correlated with $W_i$, then (4.1) will provide a much more accurate characterization of the distribution of $W_i$ than (3.1). In practice we do not specify $p_{S_i|V_i}(m|v_i)$ directly, but rather specify $p_{V_i|S_i}(v|m)$ and apply Bayes rule.\(^1\)

$$ p_{S_i|V_i}(m|v_i) = \frac{p_{S_i}(m) p_{V_i|S_i}(v_i|m)}{\sum_{m=1}^{M} p_{S_i}(m) p_{V_i|S_i}(v_i|m)}. \quad (4.2) $$

Defining $\epsilon_{i,m} \equiv p_{S_i}(m)$ and $\alpha_{i,v,m} \equiv p_{V_i|S_i}(v_i|m)$, the context-based HMM (CHMM) is parameterized by the vector $\Theta = \{\mu_{i,m}, \sigma_{i,m}^2, \epsilon_{i,m}, \alpha_{i,v,m}\}$. Given an observation of wavelet data $w$, we estimate $\Theta$ using the EM algorithm below. When only a single signal observation is available, we make the standard assumption that the wavelet coefficients in each scale are identically distributed.

\(^1\)Here, we assume that the context is discrete-valued. We can model a continuous-valued $V_i$ as an $M$-component Gaussian mixture of its own, replacing $p_{V_i|S_i}(v_i|m)$ in (4.2) with $f_{V_i|S_i}(v_i|m)$. We will find this useful in Section 4.3.
4.2 EM Algorithm for CHMMs

Recall from Section (3.1) that $J(i)$ is the scale associated with index $i$. We can then write
the EM algorithm assuming that all coefficients within a common scale are tied together
(see Section 3.5.3). Multiple signal observations, multiple wavelet trees, and models for
the scaling coefficients can be handled as outlined in Chapter 3 and in Appendix A.

Initialize: Choose $\Theta^0$ and set $I = 0$.

Expectation (E): Given $\Theta^I$, calculate (Bayes rule)

$$
p_{S_i|V_i, W_i}(m|v_i, w_i) = \frac{\epsilon_{i,m} \alpha_{i,v_i,m} f_{W_i|S_i}(w_i|m)}{\sum_{m=1}^{M} \epsilon_{i,m} \alpha_{i,v_i,m} f_{W_i|S_i}(w_i|m)}. \]

Maximization (M): Compute the elements of $\Theta^{I+1}$

$$
\epsilon_{i,m} = \sum_{k \text{ s.t. } J(k) = J(i)} p_{S_i|V_k, W_k}(m|v_k, w_k),
$$

$$
\mu_{i,m} = \frac{1}{2J(i)\epsilon_{i,m}} \sum_{k \text{ s.t. } J(k) = J(i)} w_k p_{S_k|V_k, W_k}(m|v_k, w_k),
$$

$$
\sigma^2_{i,m} = \frac{1}{2L-J(i)\epsilon_{i,m}} \sum_{k \text{ s.t. } J(k) = J(i)} (w_k - \mu_{i,m})^2 p_{S_k|V_k, W_k}(m|v_k, w_k),
$$

$$
\alpha_{i,v_i,m} = \frac{1}{\epsilon_{i,m}} \sum_{k \text{ s.t. } J(k) = J(i), v_k = v_i} p_{S_i|V_k, W_k}(m|v_k, w_k).
$$

Iterate: Increment $I \rightarrow I + 1$. Apply E and M until converged.

In contrast to the HMT E step of Appendix A, the CHMM E step is very straightforward. To ensure fast and robust training, we keep the number of free parameters in each context vector to a minimum.

4.3 Applications

To illustrate the flexibility of the CHMM framework, we now apply these models to two distinctly different problems: signal denoising and synthesis of long-range-dependent data network traffic.
4.3.1 Signal estimation

DWT methods have proved remarkably successful for estimating signals corrupted by additive white Gaussian noise (WGN) [12, 29]. The superior results of HMT model denoising have demonstrated that significant performance gains can be achieved by exploiting dependencies between wavelet coefficients. Using a CHMM, we seek similar gains, but with reduced complexity.

Since the orthogonal DWT of zero-mean WGN is again zero-mean WGN of the same power, the signal estimation problem can be posed in the wavelet domain as: Estimate the wavelet coefficients $y_i$ of a signal given the noisy measurements $w_i = y_i + n_i$, with \( \{n_i\} \) a WGN process of variance $\sigma_n^2$. As in Chapter 3, we adopt an "empirical" Bayesian approach and model the signal wavelet coefficients $Y_i$ using a two-component Gaussian mixture ($M = 2$) with $\mu_{i,1} = \mu_{i,2} = 0$.

If we knew the hidden state $S_i$ of $Y_i$, then the minimum-mean-squared-error (MMSE) estimate would be the conditional mean estimate of a Gaussian signal in Gaussian noise

$$\mathbb{E}[Y_i|w_i, S_i = m] = \frac{\sigma_{i,m}^2}{\sigma_{i,m}^2 + \sigma_n^2} w_i.$$  \hspace{1cm} (4.3)

Given probability estimates for the hidden states $S_i$, we estimate $Y_i$ as the conditional mean

$$\mathbb{E}[Y_i|w_i, \nu_i] = \sum_{m=1}^{2} p_{S_i|w_i, \nu_i}(m|w_i, \nu_i) \mathbb{E}[Y_i|w_i, S_i = m].$$  \hspace{1cm} (4.4)

If $Y_i$ is a mixture of zero-mean Gaussians, then $W_i$ is also a mixture of zero-mean Gaussians — the addition of zero-mean independent Gaussian noise increases the variance of each mixture component by $\sigma_n^2$, but leaves the state $S_i$ unaffected. Hence, we train our CHMM on the noisy wavelet data $W$ to estimate the hidden state probabilities of the signal $p_{S_i|w_i, \nu_i}(m|w_i, \nu_i)$ and (by subtracting $\sigma_n^2$) the signal mixture variances $\sigma_{i,m}^2$. We then calculate the estimates (4.4) and invert the DWT to obtain the denoised signal.

What remains is to specify contexts that are simple, yet effective, for gleaning information on the hidden states. Two simple discrete contexts that exploit clustering of signal energy in the wavelet domain illustrate our approach.
But first, we need to add some notation. Define $q_i$ as the quantized value of the wavelet coefficient $w_i$: Set $q_i = 1$ if $|w_i|^2$ is greater than the average energy in its scale, otherwise, set $q_i = 0$. Let indices $l(i)$ and $r(i)$ represent, respectively, the adjacent indices to the left and right of $i$; indices $cl(i)$ and $cr(i)$, represent, respectively, the left and right children of $i$. Also, recall from Section 2.3 that $\rho(i)$ represents the parent index of $i$.

The first context contains quantized values of the neighboring wavelet coefficients

$$V_i^{(1)} = [q_{\rho(i)}, q_{l(i)}, q_{r(i)}, q_{cl(i)}, q_{cr(i)}], \quad (4.5)$$

and thus conveys gross information about the size of the neighboring coefficients. Our intuition is that if $w_{\rho(i)}$ and $w_{cl(i)}$ are large, then there is a good chance that $w_i$ will be large as well. To encode such information ("large" vs. "small"), even crudely quantized information is sufficient. The second context combines elements of $V_i^{(1)}$ using logical or operations "\lor"

$$V_i^{(2)} = [q_{\rho(i)}, q_{l(i)} \lor q_{r(i)}, q_{cl(i)} \lor q_{cr(i)}]. \quad (4.6)$$

To further reduce complexity, we also assume that the context probabilities factor as $p_{V_i|s_i}(v_i|m) = \prod_{j=1}^{P} p_{V_{i,j}|s_i}(v_{i,j}|m)$.

In Table 3.1, we provide the MSE results for denoising Donoho and Johnstone's standard test signals [29] using CHMMs versus other state-of-the-art algorithms. Contexts 1 and 2 correspond to our proposed algorithm using the contexts defined in (4.5) and (4.6), respectively.

The key benchmarks for comparison are the IM and HMT models from Chapter 3. IM denoising employs a mixture model that treats the signal wavelet coefficients as independent. Improvements over IM signify the context's ability to capture and exploit dependencies between coefficients. Overall, the MSE performance of the context-based approach is roughly comparable to the considerably more complicated HMT denoiser of Chapter 3.
4.3.2 Signal synthesis

To further illustrate the power of the context approach, we look to the problem of modeling nonGaussian LRD network traffic. The utility and difficulty of this task are developed more thoroughly in the Chapter 1 and Chapter 5. Suffice it to say that formulating accurate models for LRD processes has been problematic, let alone nonGaussian ones.

Our goal is to demonstrate a fast wavelet-based model that is consistent both with the long-range dependence and the positive, nonGaussian marginal statistics of network traffic. Our approach will be to first train a CHMM on an actual traffic trace, and then synthesize artificial traffic with "equivalent" statistical properties. By characterizing how the wavelet coefficient variances change with scale, CHMMs can approximate the long-range dependence properties of the data. By using the Haar scaling coefficients as contexts, CHMMs can capture the positive, nonGaussian marginal properties of the traffic as we will show.

Recall from Section 2.2.3, that for the Haar DWT of nonnegative data we have $U_i > 0$ and $|W_i| \leq U_i$ (equivalently $U_{j,k} > 0$ and $|W_{j,k}| \leq U_{j,k}$ in the notation of that section). Because of this clear dependence between $W_i$ and $U_i$, we use the random variable $V_i = U_i$ as the context for the random variable $W_i$. This is equivalent to modeling the pair $(U_i, W_i)$ as a two-dimensional mixture of Gaussians with diagonal covariance matrices of the form

$$
\begin{bmatrix}
\tilde{\sigma}_{i,m}^2 & 0 \\
0 & \sigma_{i,m}^2
\end{bmatrix}.
$$

The parameters $\tilde{\mu}_{i,m}$ and $\sigma_{i,m}^2$ can be updated in the M step in a fashion similar to the updates for $\mu_{i,m}, \sigma_{i,m}^2$.

Thus, this procedure employs a mixture model to approximate the 2-d density for $(U_i, W_i)$ and then uses the 2-d density to obtain a conditional density for $W_i$ based on $U_i$. With enough mixture parameters, this approach in theory can approximate $(U_i, W_i)$ to arbitrary precision, hence automatically learning the constraints $U_i > 0$ and $|W_i| < U_i$. However, this would require a huge number of mixtures and is impractical.

To simplify our modeling, we map the cone $U_i > 0, |W_i| < U_i$ to the plane through the invertible map $h : (U_i, W_i) \mapsto \left( \log(U_i), -\text{sgn}(W_i) \log(1 - |W_i|/U_i) \right)$. By modeling $h(U_i, W_i)$ and then inverting to form $(U_i, W_i)$, we automatically enforce the positivity constraints. To synthesize $W_i$ given $U_i$, we map $U_i$ to $\log(U_i)$, use it as a context to synthesize
the transformed data, generate a realization, and then invert the map $g$ to produce $W_t$.

To synthesize an entire wavelet transform $W$, we work in "top-down" fashion starting from the root of the wavelet tree by synthesizing the single coarsest scale wavelet coefficient. (We assume its context, the global mean of the signal, is already specified.) We iterate down the tree using the fact that summing and differencing $U_i$ and $W_i$ provides the context information for synthesizing $W_{ct(i)}$ and $W_{cr(i)}$.

As a test, we trained the CHMM synthesis algorithm on a portion of the Bellcore Ethernet (BC-pAug89) trace [54], namely the first $10^6$ interarrival times of the day-long trace started August 29, 1989. (See Chapter 5 for a more in-depth discussion and analysis of this data.) The model was equipped with ten mixture-components ($M = 10$) at each wavelet scale. Ten was the minimum number of mixtures needed to match the trimodal behavior of the data. This behavior is illustrated in Figure 4.2, where we compare, over different time scales, a random realization from our synthesis algorithm with the actual data. In Figure 4.3, we illustrate the histogram fit that our synthesis algorithm achieves over different time scales. Note that by fitting a high-dimensional mixture model or HMM directly to the time-domain data, one could obtain a good match to the first-order density. However, then the LRD properties would be lost. Similarly, a dfGn process could be used to approximately match the LRD properties, but then the marginal density would be Gaussian, quite different from the heavy-tailed data histogram.

As is evident from the Figures, CHMM synthesis captures both the marginal properties of the traffic and, because of the match over a number of time scales, the long-range dependence as well. Though applied in the wavelet-domain, the CHMM data leads to a multimodal time-domain marginal quite close to that of the original data. The major difference is that the CHMM histogram is a bit smoother than the actual data, which we would expect with any kind of parametric fit. In modeling this trimodal behavior of the data, we believe the large number of mixture parameters is necessary only at the finest scales — at the coarsest scales the data is unimodal, which suggests that the number of mixtures parameters may be reduced.
Figure 4.2: The Bellcore BC-pAug89 interarrival times and a synthetic trace from the CHMM. Shown are interarrival times of groups of packets (top) one-hundred, (middle) ten, and (bottom) one packet from the start of each trace.

Figure 4.3: Histograms of the interarrival times corresponding to the data from Figure 4.2 for groups of (top) one-hundred, (middle) ten, and (bottom) one packet.
4.4 Discussion

CHMMs have a number of potential advantages over conventional HMMs for exploiting the wavelet-domain structure inherent in real-world signals. First, CHMMs allow the user to characterize dependencies that may be too complex or even downright impossible to model using standard HMMs. Second, although efficient algorithms exist for HMMs based on trees, for more complicated graph structures (such as 2-d HMMs for images), the training procedure can become intractable. CHMMs deal naturally with noncausal information, yet retain the simplicity of a causal model. The explanation lies in the fact a CHMM consists essentially of a series of local models, each with a small number of parameters, that can be trained independently. More traditional HMM models, on the other hand, adjust their parameters to optimize a complicated global objective function.

The primary disadvantage of the CHMM framework is that it lacks the feedback mechanism of more traditional HMMs that allow the model to propagate information from variables across the entire model, hence capturing dependencies from more than just neighboring wavelet coefficients. However, in many instances, we expect the convenience and efficiency of the context approach to outweigh this potential limitation.
Chapter 5

The Multifractal Wavelet Model With Application To Network Traffic

Recall from Section 2.2 that the wavelet transform serves as an approximate Karhunen-Loéve transform for $1/f$ and LRD processes that are otherwise difficult to model [38, 48]. Gaussian LRD processes can be approximately synthesized by generating wavelet coefficients as independent zero-mean Gaussian random variables, identically distributed within scale according to $W_{j,k} \sim N(0, \sigma_j^2)$, with $\sigma_j^2$ chosen to match the measured or theoretical variances of the wavelet coefficients of the desired process [59]. We call the resulting model the wavelet-domain independent Gaussian (WIG) model [59].

However, for important applications such as networking and finance, the LRD data are inherently-positive valued and oftentimes characterized via skewed, heavy-tailed distributions. Until now, wavelet-domain modeling of such processes has assumed Gaussian-distributed wavelet coefficients and hence an unrealistic assumption of an underlying Gaussian signal.

For these applications of interest, any theoretical results or simulations derived via a model producing negative data are at best questionable and at worst nonsensical. Thus, it is crucial that the wavelet-domain models preserve key time-domain properties such as positivity. Moreover, the model should be as simple and intuitive as possible to ensure its utility for theoretical analysis and interpretation.

The CHMM model developed in Chapter 4 can model an observed set of positive LRD data quite admirably. However, the CHMM is too complex to analyze and apply in a more general setting. For instance, there is no clear relationship between the CHMM mixture parameters and the Hurst parameter $H$. Thus, in terms of the five secondary objectives of
this thesis in Section 1.2, the CHMM achieves the first but, due to its lack of interpretability, fails to achieve the rest.

To strike a balance between the simplicity of the WIG and the accuracy of the CHMM, we stick with the Haar wavelet transform but consider a simple multiplicative construction that we call the Multifractal Wavelet Model (MWM). Although the MWM can be understood on a stand-alone basis, the MWM is akin to a binomial multifractal, and as such, can be analyzed within an abstract multifractal framework.

For a practical application of the MWM, we investigate further the increasingly vital area of data networking. We demonstrate that the MWM fits actual network traffic more closely than current state-of-the-art models such as the WIG [89].

5.1 Notation

In this chapter we seek develop a wavelet-domain model for a positive, stationary, LRD signal \( C(t) \) and its integral \( D(t) \). We use the terminology \( C \) to denote a nonnegative signal to be modeled using the MWM (as opposed to the pedestrian signal \( X \), which could be negative). \( D(t) \), though not crucial in this work, is the signal of interest for a theoretical multifractal analysis [89], since at infinite resolution \( C(t) \) may only be defined in a distributional sense.

In practice, we will work with a discrete-time signal \( C^{(L)}[k] \) that approximates \( C(t) \) at resolution \( 2^{-L} \). Using the Haar wavelet, the discrete process \( C^{(L)}[k] \) takes values that correspond to the integral of \( C(t) \) in the interval \([k2^{-L}, (k+1)2^{-L})\). Such processes have a natural interpretation as an increment process:

\[
C^{(L)}[k] \equiv D\left((k+1)2^{-L}\right) - D\left(k2^{-L}\right) = \int_{k2^{-L}}^{(k+1)2^{-L}} C(t) \, dt = 2^{-L/2} U_{L,k} \quad (5.1)
\]

for \( k = 0, \ldots, 2^L - 1 \).

This corresponds to a single scaling coefficient tree approximating \( C(t) \) on the interval \([0, 1]\). While we will emphasize this case in the sequel, in certain cases (as in Section 5.3.4 below), we will find it convenient to employ a forest of \( R \) trees rooted at \( R \) scaling
coefficients $U_{0,k}, k = 0, 1, \ldots, R - 1$. This goes back to notion of the wavelet transform as a forest of binary trees (see Section 3.1). In this case, the process $C(t)$ is assumed to lie in the interval $[0, R]$.

5.2 Multifractal Wavelet Model

To be useful in real applications, our model must be simple, produce a fast analysis and synthesis, and closely match the process's positive, non-Gaussian marginals and its LRD. We will now show how this is possible using a simple Haar wavelet construction of the increments process $C^{(I)}[k]$.

5.2.1 Positivity through multiplication

We wish to build a statistical model for the $W_{j,k}$'s that automatically incorporates (2.36). This leads us to a simple multiplicative signal model. Let $A_{j,k}$ be a random variable supported on the interval $[-1, 1]$ and define the wavelet coefficients by

$$W_{j,k} = A_{j,k} U_{j,k}.$$  \hspace{1cm} (5.2)

In Section 5.2.4 we will place some additional constraints on the $A_{j,k}$. The multifractal wavelet model (MWM) consists of the Haar wavelet transform and the structure constraint (5.2).

5.2.2 Synthesis procedure

The MWM can be interpreted as a simple coarse-to-fine synthesis running as follows (see Figure 5.1):

1. Set $j = 0$. Fix or compute the coarsest (root) scaling coefficient $U_{0,0}$ (modeling of $U_{0,0}$ is discussed in Section 5.3.4).

2. At scale $j$, generate the random multipliers $A_{j,k}$ and calculate each $W_{j,k}$ via (5.2) for $k = 0, \ldots, 2^j - 1$. 
Figure 5.1: (a) More detailed tree structure of scaling coefficients. (b) MWM construction: At scale $j$, we form the wavelet coefficient as the product $W_{j,k} = A_{j,k} U_{j,k}$, with $A_{j,k}$ a random variable distributed in $[-1, 1]$. Then, at scale $j + 1$, we form the scaling coefficients $U_{j+1,2k}$ and $U_{j+1,2k+1}$ as sums and differences of $U_{j,k}$ and $W_{j,k}$ (normalized by $1/\sqrt{2}$).

3. At scale $j$, use $U_{j,k}$ and $W_{j,k}$ in (2.35) to calculate $U_{j+1,2k}$ and $U_{j+1,2k+1}$, the scaling coefficients at scale $j + 1$ for $k = 0, \ldots, 2^j - 1$.

4. Iterate steps 2 and 3, replacing $j$ by $j + 1$ until the scale $j = L$ is reached.

Since we generate the scaling coefficients simultaneously with the wavelet coefficients, there is no need to invert the wavelet transform. The finest-scale scaling coefficients are in fact the MWM output process, i.e., $C^{(L)}[k] = 2^{-L/2} U_{L,k}$, $k = 0, \ldots, 2^L - 1$. The total cost for computing $N$ MWM signal samples is $O(N)$.

Because of the simple structure of the Haar transform, Steps 2 and 3 above can be combined, eliminating the wavelet coefficients altogether:

$$U_{j+1,2k} = \left( \frac{1 + A_{j,k}}{\sqrt{2}} \right) U_{j,k} \quad \text{and} \quad U_{j+1,2k+1} = \left( \frac{1 - A_{j,k}}{\sqrt{2}} \right) U_{j,k}. \quad (5.3)$$

5.2.3 Closed-form coefficient expressions

Because of its simplicity, we can easily obtain explicit formulas for the MWM's fine-scale Haar wavelet and scaling coefficients in terms of the scaling coefficients and multipliers at
coarser scales. We begin by defining an indexing scheme to relate the coarsest-scale scaling coefficient $U_{0,0}$ to its "descendants" at finer scales, the scaling coefficients $U_{j,k}$, $j > 0$ (see Figure 5.1(a)). Let $k_j$, $j > 0$, be the variable indexing the possible shifts of the descendants of $U_{0,0}$ at scale $j$. We can relate the shift $k_j$ of a scaling coefficient to the shift of one of its two direct descendants (children) $k_{j+1}$ via $k_{j+1} = 2k_j + k'_j$, with $k'_j = 0$ corresponding to the left descendant and $k'_j = 1$ the right descendant (see Figure 5.1(a)). From this we can express $k_j$ as a binary expansion in terms of the $k'_i$ ($i = 0, \ldots, j - 1$),

$$k_j = \sum_{i=0}^{j-1} k'_i 2^{j-1-i}.$$  \hfill (5.4)

Moreover, $k_j = \left[ \frac{k_{j+1}}{2} \right]$ and $k'_j = k_{j+1} - 2 \left[ \frac{k_{j+1}}{2} \right]$, with $\lfloor x \rfloor$ the largest integer less than or equal to $x$. Note that fixing a sequence $k'_i$ specifies not only $k_j$, but a "line of descendants" of $U_{i,k_i}$ ($i = 0, \ldots, j$) from $U_{0,0}$ down to $U_{j,k_j}$. Using this notation, we can derive closed-form expressions for the MWM wavelet and scaling coefficients.

**Proposition 1** Define the wavelet coefficients of the Haar wavelet system through (5.2), with the random variables $A_{j,k}$ supported on $[-1, 1]$. We then have the general relations

$$U_{j,k_j} = 2^{-j/2} U_{0,0} \prod_{i=0}^{j-1} \left[ 1 + (-1)^{k'_i} A_{i,k_i} \right]$$ \hfill (5.5)

and

$$W_{j,k_j} = 2^{-j/2} A_{j,k_j} U_{0,0} \prod_{i=0}^{j-1} \left[ 1 + (-1)^{k'_i} A_{i,k_i} \right].$$ \hfill (5.6)

**5.2.4 Properties of the MWM**

**Additional constraints on the multipliers**

The Haar wavelet coefficients of a stationary signal will be, using (2.26), identically distributed within each scale with $E[W_{j,k}] = 0$. To model these properties in the MWM, we will assume that, within each scale $j$, we have the following:
1. the multipliers $A_{j,k}, \ k = 0, 1, \ldots, 2^j-1$, are identically distributed according to some random variable $A_{(j)} \in [-1, 1],$

2. the $A_{(j)}$ are symmetric about 0, and

3. (simplifying assumption) the $A_{j,k}$ are independent of both the coarsest scaling coefficient $U_{0,0}$ and the $A_{l,k}$ on finer scales $l > j$.\(^1\)

**Marginal density and stationarity**

Under the above assumptions, Proposition 1 leads us to the marginal density and stationarity properties of $C^{(L)}[k]$. Setting $j = L$ in (5.4) and (5.5), and setting $k = k_L$ in (5.1) yields\(^2\)

$$C^{(L)}[k] = 2^{-L} U_{0,0} \prod_{j=0}^{L-1} \left( 1 + (-1)^j A_{j,k_j} \right) \overset{d}{=} 2^{-L} U_{0,0} \prod_{j=0}^{L-1} (1 + A_{(j)}). \quad (5.7)$$

Thus, $C^{(L)}[k]$ is first-order stationary and identically distributed. Note that without the requirement that $A_{(j)}$ be symmetric, the marginal distribution of $C^{(L)}[k]$ would depend on $k$ and (5.7) would not hold. Hence, symmetry of the multipliers is key for modeling stationary processes.

However, $C^{(L)}[k]$ will not be second-order stationary in general. Due to the dyadic structure of the wavelet transform, wide-sense stationarity of $C^{(L)}[k]$ is unattainable using a wavelet-domain model with uncorrelated wavelet coefficients (except in the trivial case of white noise). In the MWM, for a fixed shift $m$, $\mathbb{E}[C^{(L)}[k + m]C^{(L)}[k]]$ will vary as a function of $k$ in relation to the size of the smallest subtree containing both $C^{(L)}[k + m]$ and $C^{(L)}[k]$. If the $A_{j,k}$ multipliers are independent and identically distributed (iid), then the smaller the subtree, the stronger the potential correlation.

---

\(^1\)Strictly speaking, for our development we need only assume independence along “lines of descendants.” That is, multipliers on different scales can be dependent as long as one is not a descendant of the other.

\(^2\)The symbol “$\overset{d}{=}$” denotes equality in distribution
Given our independence assumptions, the moments of \( C^{(L)}[k] \) are readily calculable from (5.7) via

\[
\mathbb{E}[C^{(L)}[k]^q] = \mathbb{E}[U_{0,0}] \prod_{j=0}^{L-1} \mathbb{E} \left[ \left( \frac{1 + A(j)}{2} \right)^q \right].
\] (5.8)

As we increase the number of scales in the wavelet transform \( (L \to \infty) \), an appropriately scaled version of \( C^{(L)}[k] \) converges to a log-normal random variable as long as \( \mathbb{E} \left[ \log(A(j))^3 \right] \) is bounded for \( j \geq 0 \). This follows from the application to \( \log(C^{(L)}[k]) \) of the Berry-Esseen theorem [76], a Central Limit Theorem for non-identically distributed random variables.

**Wavelet-domain dependency structure**

If we assume that the \( A_{j,k} \)'s are independent both between scales and within scales, then the wavelet coefficients will be dependent, but uncorrelated. This lack of correlation follows from the fact that terms of the form \( \mathbb{E}[A_{j,k}] \) factor out of any correlation calculation, with \( \mathbb{E}[A_{j,k}] = 0 \). However, a higher-order dependency structure remains, which is of course key for preserving signal positivity.

While a dependency structure with no correlations between wavelet coefficients may at first seem somewhat unnatural, such models are not entirely unrealistic. Wavelet coefficients of random signals can exhibit minimal second-order correlations (approximately decorrelated via the Karhunen-Loève transform), yet still have strong dependencies in higher-order moments. For instance, many real-world data sets exhibit strong dependencies in the energy of the wavelet coefficients, corresponding to fourth-order cross-moments [20, 75].
5.2.5 The MWM is a multifractal

Recall from Section (2.4.2) the construction of the binomial cascade using multipliers $M_i^k$. Let us try to equate the MWM with this construction. If we set $M_0^0 = U_{0,0}$ and

$$M_{k_j}^j = \frac{(1 + (-1)^{k_j-1} A_{j-1,k_j-1})}{2},$$

then by comparing (5.7) with (2.44), we see that the MWM is a random binomial cascade.

Recall from Section 5.2.3 that $k_{j-1}' = k_j - 2 \left\lfloor \frac{k_j}{2} \right\rfloor$ and $k_{j-1} = \left\lfloor \frac{k_j}{2} \right\rfloor$. Substituting in $k_j = 2n$ and $k_j = 2n + 1$ into (5.9) and applying these definitions, we obtain

$$M_{2n}^j + M_{2n+1}^j = \frac{1 + A_{j-1,n}}{2} + \frac{1 - A_{j-1,n}}{2} = 1.$$

Hence, the MWM is a conservative cascade.

A little more work shows that the MWM, as we will apply it, satisfies the conditions to be a multifractal [89]. Identifying the MWM algorithm with cascades and multifractals allows us to benefit from the accumulated knowledge of the field of multifractals, including a precise understanding of the convergence of the algorithm, properties of the marginal distributions, and a range of possible refinements and extensions [89]. The MWM corresponds to a dyadic, conservative binomial cascade, but multifractal theory allows for non-dyadic non-conservative cascades. Such cascades could better model stationarity and other signal properties.

5.2.6 Related work

Constructions similar to the MWM were developed earlier in [18, 77]. A similar multiplicative model for wavelet coefficients has been developed in [102, 103], where it is applied to wavelet-domain Bayesian estimation of the intensity of a Poisson process. There, the $A_{j,k'}$'s are independent multipliers that, within each scale, are identically-distributed as mixtures of $\beta$ random variables. The primary difference with this work is that we model the data directly, whereas [102, 103] models a wavelet-domain prior density for the intensity function of a Poisson process.
In Chapter 4 we modeled the wavelet coefficients using a CHMM. It can be shown that this model corresponds to (5.2), again with the \( A_{j,k} \)'s identically-distributed within each scale, but with each \( A_{j,k} \) distributed according to a mixture density dependent on the value of \( U_{j,k} \). Although this model proves to be quite flexible and accurate for characterizing positive LRD data, it requires iterative, maximum-likelihood (expectation-maximization) training and is difficult to characterize analytically.

## 5.3 Data Modeling Using the MWM

To complete our model, we now specify probability density functions (pdfs) for the coarsest scaling coefficient \( U_{0,0} \) and for the \( A_{(j)} \) multipliers at each scale. We can use the degrees of freedom in these pdfs in order to control two key signal properties. First, we control the covariances and LRD of the output signal \( C^{(L)}[k] \) through the wavelet energy decay. Second, we control the higher-order moments and marginal pdf of \( C^{(L)}[k] \) through the scaling coefficient moments.

### 5.3.1 Controlling the wavelet energy decay

**Empirical approach**

To approximate the covariance behavior of a target signal, we vary the wavelet energy decay across scale. We choose the pdfs for the \( A_{(j)} \)'s to control the wavelet coefficients’ scaling behavior via (5.6). The fact that this scaling behavior allows us to model covariances can be explained as follows.

Consider the Karhunen-Loève properties of the wavelet transform. Previous work \([38, 48, 110]\) has demonstrated that the wavelet transform approximately decorrelates or whitens a general class of LRD signals. If the decorrelation were exact, then specifying the correct variances of the wavelet coefficients would fully capture the covariance structure of the signal. Since this decorrelation is approximate, we can approximately control the covariance behavior by appropriately setting the second moments (energies) of the wavelet
coefficients at each scale.

The simplest way to control energy scaling is to fix the energy at the coarsest scale $(j = 0)$ and then set the ratios of energy for the other scales with $\eta_j = \frac{\text{var}(W_{j-1,k})}{\text{var}(W_{j,k})}$, $0 \leq j < n$. For a stationary $1/f$ process, we see from (2.29) that $\eta_j = 2^{2H-1}$ is constant. Using Proposition 1 we can calculate the $\eta_j$'s of the MWM via

$$
\eta_j = \frac{\mathbb{E}[W_{j-1,k}^2]}{\mathbb{E}[W_{j,k}^2]} = \frac{2 \mathbb{E}[A_{U-1}^2] \mathbb{E}[U_{j-1,k}^2]}{\mathbb{E}[A_{U}^2] \mathbb{E}[(1 + A_{j-1})^2] \mathbb{E}[U_{j-1,k}^2]} = \frac{2 \mathbb{E}[A_{U-1}^2]}{\mathbb{E}[A_{U}^2] \left(1 + \mathbb{E}[A_{j-1}^2]\right)}.
$$

(5.11)

To match a given variance decay, we can recursively solve (5.11) for $\mathbb{E}[A_{U}^2]$ in terms of $\eta_j$ and $\mathbb{E}[A_{j-1}^2]$ for $j = 1, 2, \ldots, L - 1$. We initialize the calculation at the coarsest scale $(j = 0)$ through

$$
\mathbb{E}[A_{(0)}^2] = \frac{\mathbb{E}[W_{0,0}^2]}{\mathbb{E}[U_{0,0}^2]}.
$$

(5.12)

Analytical approach

It is possible to calculate the theoretical time-averaged covariance function for the continuous-time MWM process $C(t)$. Since the wavelet coefficients are uncorrelated across scale, the time-averaged covariance of the process may be calculated as a superposition of the time-averaged covariance functions from each scale. Since the wavelet coefficients are uncorrelated within scale, the contribution from scale $j$ to the covariance function boils down to the integral

$$
\int_{t=0}^{2^{-j}} \mathbb{E}[W_{j,0}^2 \psi_{j,0}(t) \psi_{j,0}(t + \tau)] \, dt = \mathbb{E}[W_{j,0}^2] \psi_{j,0}(t) * \psi_{j,0}(-t),
$$

(5.13)

with $*$ denoting convolution. Thus, the contribution from a given scale to the covariance function is a piecewise linear function corresponding to the deterministic autocorrelation of the Haar wavelet function at that scale.\textsuperscript{3} It is easy to show that the length-$N$ discrete-time

\textsuperscript{3}The fact that the time-averaged covariance is piecewise linear for the Haar DWT was recognized in [59, 103]. However, these works did not mention the fact that covariance is a superposition of these deterministic wavelet autocorrelations, a result which also holds for more general wavelet systems.
version \( C^{(L)}[k] \) is also piecewise linear.

Hence, in matching a process's wavelet energy decay, we are essentially fitting a length-
\( N \) covariance function as a superposition of \( L = \log_2 N \) piecewise-linear wavelet autocorrelation functions. The weights applied to these wavelet autocorrelation functions are given by the wavelet variances at each scale. Thus, we can fit an MWM model to data given only its mean and covariance function. For instance, using a least squares approach we can solve for the wavelet coefficient variances that best match a given covariance function. This analytical approach may be more accurate than the empirical approach, which can incur error in the estimation of the wavelet variances at each scale.

### 5.3.2 Controlling the moments of the scaling coefficients

It is easily shown that the moments of the scaling coefficients scale according to

\[
\frac{\mathbb{E}[U_{j-1,k}^q]}{\mathbb{E}[U_{j,k}^q]} = 2^{q/2} \mathbb{E}[(1 + A_{(j-1)})^q]^{-1}.
\]  

(5.14)

Through (5.14) we can control the scaling of the higher-order (and even negative) moments of the scaling coefficients — and thus of \( C^{(L)}[k] \) — through the moments of the \( A_{(j)} \)'s.

### 5.3.3 Distributions for the multipliers

We will investigate two distributions for the multipliers, the symmetric \( \beta \) distribution and a symmetric point-mass distribution. Both of these distributions are compactly supported, easily shaped, and amenable to closed-form calculations.

#### Symmetric beta distribution

A \( \beta(p, p) \) random variable \( A \), symmetrically distributed over \((-1, 1)\), has pdf [45]

\[
g_A(a) = \frac{(1 + a)^{p-1}(1 - a)^{p-1}}{B(p, p) 2^{2p-1}}.
\]  

(5.15)
Figure 5.2: Examples of the pliable pdf $g_A(a)$ of the $\beta(p, p)$ random variable $A$, for different values of $p$. For $p = 0.2$, $A$ resembles a binomial random variable, and for $p = 1$ it has a uniform density. For $p > 1$ the density resembles a truncated Gaussian density, with the resemblance increasing with $p$.

Here $B(\cdot, \cdot)$ is the beta function, and $p > 0$ is a shape factor (see Figure 5.2). For large $p$, the $\beta(p, p)$ approximates a Gaussian distribution [45]. The variance is given by

$$\text{var}(A) = \mathbb{E}[A^2] = \frac{1}{2p + 1}. \quad (5.16)$$

Combined with (5.11), (5.16) tells us how to choose the $p$'s to obtain the desired scaling behavior as parameterized via $\eta_j$. Denoting by $p_{(j)}$ the beta parameter at scale $j$, we find that

$$p_{(j)} = \frac{\eta_j}{2} (p_{(j-1)} + 1) - 1/2. \quad (5.17)$$

When we use $\beta$-distributed multipliers, we call the model the $\beta$ multifractal wavelet model ($\beta$MWM).

Point-mass distribution

The point mass distribution we consider is non-zero at three points

$$\Pr[A = c] = \Pr[A = -c] = r, \quad \text{and} \quad \Pr[A = 0] = 1 - 2r, \quad (5.18)$$

with $0 \leq r, c \leq 1$. Although seemingly not as rich as the $\beta$, this distribution has two parameters and thus can match an additional higher-order moment of the signal.
The point-mass distribution has variance \( \text{var}(A) = 2rc^2 \). The higher order moments of \( \left( \frac{1+A}{2} \right)^q \), which are useful for characterizing the scaling coefficient moments (see (5.8)), are given by

\[
\mathbb{E}\left[ \left( \frac{1+A}{2} \right)^q \right] = 2^{-q} r \left( (1-c)^q + (1+c)^q \right) + 2^{-q}(1-2r). \tag{5.19}
\]

### 5.3.4 Distribution for the root scaling coefficient

What remains is to model the density of \( U_{0,0} \), the root of the tree in Figure 5.1. In theory, this distribution should be strictly positive. However, if there are enough scales in the wavelet transform, we can appeal to Central Limit Theorem-type arguments (although LRD makes precise analysis somewhat cumbersome) that the root scaling coefficient is approximately Gaussian, thus characterized only through its mean \( E[U_{0,0}] \) and variance \( \text{var}(U_{0,0}) \). Crucial to this assumption is that the mean greatly outweighs the variance so that the probability of a negative value is negligible.

Although our development has focused on a single wavelet tree with a single scaling coefficient \( U_{0,0} \), in certain synthesis applications it is useful for the MWM to employ several wavelet trees with one root scaling coefficient per tree. For instance, we may wish to synthesize a trace of length \( 2^{L_0} \), but have only enough coarse-scale information to form a model over \( L < L_0 \) scales. In this case, we can concatenate \( 2^{L_0-L} \) length-\( 2^L \) traces, which corresponds to an MWM with \( 2^{L_0-L} \) iid coarsest-scale scaling coefficients \( U_{0,k} \). Of course, an iid assumption for the \( U_{0,k} \) is suboptimal in that it destroys LRD over time lags greater than \( 2^L \). This problem, along with a potential solution, is discussed further in Section 5.3.6.

### 5.3.5 Modeling positive \( 1/f \) noise

We next investigate how to parameterize the MWM in order to model a positive-valued ESS process (see Section 2.1.2) with Hurst parameter \( H \). It is easily seen from (2.29) that we should choose \( \eta_j = 2^{2H-1} \) independently of scale. This leads to:
Table 5.1: Asymptotic values for the shape $p$ and variance $\mathbb{E}(A^2)$ of the $\beta$ multipliers $A_{j,k}$ required to fit ESS noise with parameter $H$.

<table>
<thead>
<tr>
<th>$H$</th>
<th>0.55</th>
<th>0.6</th>
<th>0.65</th>
<th>0.7</th>
<th>0.75</th>
<th>0.8</th>
<th>0.85</th>
<th>0.9</th>
<th>0.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>0.077</td>
<td>0.175</td>
<td>0.301</td>
<td>0.470</td>
<td>0.707</td>
<td>1.06</td>
<td>1.66</td>
<td>2.86</td>
<td>6.47</td>
</tr>
<tr>
<td>$\mathbb{E}[A^2]$</td>
<td>0.866</td>
<td>0.741</td>
<td>0.625</td>
<td>0.516</td>
<td>0.414</td>
<td>0.320</td>
<td>0.231</td>
<td>0.149</td>
<td>0.072</td>
</tr>
</tbody>
</table>

**Proposition 2** Assume that the $A_{j,k}$ in (5.2) are iid within each scale $j$ (distributed as $A_{(j)}$), supported on $[-1, 1]$, symmetric about 0, and such that

$$
\mathbb{E}[A_{(j)}^2] = \frac{2^{2-2H} \mathbb{E}[A_{(j-1)}^2]}{1 + \mathbb{E}[A_{(j-1)}^2]}.
$$

(5.20)

Then the MWM output process $C^{(L)}[k] = 2^{-L/2} U_{L,k}$ is positive and exhibits power-law behavior of the wavelet coefficient energies (2.30) with exponent $2H - 1$. Moreover,

$$
\lim_{j \to \infty} \mathbb{E}[A_{(j)}^2] = 2^{2-2H} - 1, \quad 1/2 < H < 1.
$$

(5.21)

The first part, i.e. (5.20), follows from (5.11). By solving (5.20) for the fixed-point, we obtain (5.21). A simple analysis of (5.21) shows that for $1/2 < H < 1$ the iteration is well-defined on all scales, since the variance of $A_{(j)}$ must lie in $[0, 1]$ for all $j$.

If we use a $\beta$ distribution for the multipliers, the fixed point formula for the variance $\mathbb{E}[A^2]$ leads to a fixed point for $p$ of the form

$$
p = \lim_{j \to \infty} p_{(j)} = \frac{2^{2H-1} - 1}{2 - 2^{2H-1}}, \quad 1/2 < H < 1.
$$

(5.22)

Table 5.1 provides typical fixed-point values for $p$ and the variance $\mathbb{E}[A^2]$ given the desired $H$. There is no such expression for the point-mass distribution, since even though the variance converges, an extra degree of freedom remains available for matching higher-order moments. We conclude that the MWM can approximate a positive-valued ESS process with Hurst parameter $1/2 < H < 1$ to infinitely fine resolution.
5.3.6 Fitting the MWM to data measurements

We now develop a procedure for fitting the MWM to actual data measurements. The first step in the fitting is a wavelet analysis: we compute the wavelet coefficients of the measurements (a length-$N$ signal) using a Haar wavelet transform algorithm (filter bank, etc. [21, 107]). The number of wavelet scales in the transform, $n$, is chosen as mentioned below.

We require $\text{var}(W_{j,k})$, $j = 0, \ldots, L - 1$ and $\mathbb{E}[U_{0,0}^2]$ to fit the MWM via (5.11) and (5.12). (Values for the higher-order scaling coefficient moments (5.14) may also be useful if the multiplier densities have more than one free parameter.) There exist two reasonable approaches for selecting these values. We can either plug in the empirical wavelet variances directly, or we can assume a parametric model for the variances and use the measured data to fit the model.

If we plug the empirical moments directly into (5.11) and (5.12), we must ensure that we have enough data to collect reliable statistics. This problem is most pressing for the coarsest-scale wavelet and scaling coefficients, of which we have the fewest. In practice, we set the number of levels $n$ of the Haar transform such that the number $\lfloor N2^{-L} \rfloor$ of coarsest-scale wavelet and scaling coefficients is sufficient for estimating $\mathbb{E}[W_{0,0}^2]$ and $\mathbb{E}[U_{0,0}^2]$.

A parametric model for the moment scaling would allow us to extrapolate the coarse-scale scaling and wavelet coefficient moments that we have difficulty measuring due to lack of data. It would also render the modeling more robust and provide a more concise representation of the data's behavior. Parametric models for $\eta_j$ as a function of scale are currently under investigation.

In some cases, it may be impossible to exactly match the moment scaling of the data using the MWM. The scaling of moments of the actual data may be inconsistent with the possible moments of the $A_{j,k}$ multipliers. For instance, the positive moments of $A_{j,k}$ are bounded above by those of a random variable with point masses of weight $1/2$ at $-1$ and at $1$. The moment scaling of certain data may lead to multiplier moment constraints outside these bounds that cannot be fit exactly. This could occur, for example, if the data exhibited
dependencies between the $A_{j,k}$ and $U_{j,k}$.

5.4 Application to Network Traffic

To demonstrate the flexibility of the MWM, we will separately apply it to fit interarrival times and bytes-per-time traces. The first experiment, applied to interarrival times, will involve an exploratory approach to introduce us to the basic issues involved with modeling LRD and multifractal data. The second experiment, applied to bytes-per-time, will provide a comparison between the MWM model and the current state-of-the-art WIG model. Recall that the WIG is similar to the dfGn, but can model a wider range of LRD processes.

5.4.1 Modeling interarrival times

The first real data set is one of the famous Ethernet data traces collected at Bellcore Morristown Research and Engineering facility [54]. The trace (BC-pAug89) began at 11:25 on August 29, 1989, and ran for about 3142.82 seconds (until 1,000,000 packets had been captured). We focus on the $10^6$ interarrival times of Figure 5.3(a). Although slightly dated, this data set provides a well-known benchmark useful for examining the fractality and LRD of network traffic.

First, we analyze the properties of the BC-pAug89 trace. We use the variance-time plot (Figure 5.4) to obtain a qualitative characterization of the covariances present in the data. From the plot, we find the trace exhibits LRD with $H \approx 0.79$. Since the plot is somewhat "kinked," the trace most likely does not exhibit a strict second-order scaling. As Figure 5.3 plainly shows, modeling BC-pAug89 as an dfGn process with $H = 0.79$ and the same mean and variance leads to nearly 30% of the synthesized data being negative. The culprit is the large standard deviation to mean ratio of 1.8 of BC-pAug89. The oft used but ad hoc procedure of setting all negative points to zero would clearly result in a process with very different statistics to those required. In general, dfGn models are of limited utility for positive data with small mean and large variance.
Figure 5.3: Interarrival times of groups of packets of (a) Bellcore August 1989 sBC-pAug89 data [54], (b) one realization of the multifractal wavelet model (MWM) synthesis, and (c) one realization of discrete fractional Gaussian noise (dfGn) synthesis. The top, middle, and bottom plots correspond to interarrivals of one-hundred packets, ten packets, and one packet, respectively. The ten-packet and one-packet plots correspond to the last tenth of the data from the one-hundred-packet and ten-packet plots, respectively, as indicated by the vertical dotted lines. Approximately 30% of the dfGn values are negative.

Moving beyond second-order statistics, we measure the multifractal properties of BC-pAug89. As discussed in Section 2.4.2, we can estimate the partition $T(q)$ (2.39) as the slope of a linear fit of the log-log plot of the sample moments. In Figure 5.5(a) the only noticeable deviation from linearity is at the very finest resolution of analysis — a fact that
Figure 5.4: Variance-time plot of the Bellcore BC-pAug89 data "x" and one realization of the βMWM synthesis "o". Here m denotes the level of aggregation defined through (2.18).

is enhanced in Figure 5.5(b), where the increments of the log-log plot are displayed. With 16 octaves (5 decades) of excellent scaling, we can be confident in concluding that BC-pAug89 is multifractal.

Figure 5.5: (a) Scaling of log-moments $\log_2 (E_j(q))$ (2.40) vs. scale j for the $10^6$ Bellcore inter-arrival times BC-pAug89 with q ranging from $-3.2$ to $3.2$ and j ranging from 1 to 19, with $j = 19$ the finest scale. (b) Increments of the log-log scaling shown in (a). The closeness of the linear fits in (a), as indicated by the stable behavior of the increments in (b), indicates that the interarrival times are indeed multifractal.

Applying the Legendre transform, we obtain the multifractal spectrum (2.41) of Figure 5.6. This function gives the large deviations from the "most frequent" singularity ex-
Figure 5.6: Multifractal spectra (2.41) of the Bellcore BC-pAug89 data, $\beta$MWM synthesis, and a hybrid MWM employing beta distributions at coarse scales and point masses at fine scales. The spectra were obtained through the Legendre transform of the scaling of the moments (see Figures 5.5 and 5.7). The close match in the upper left part, which corresponds to $q$ values (=slopes of tangents to the spectrum) between 0 and 2, indicates that the $\beta$MWM matches these low ($q_{th}$) order moments very well. The divergence of the spectra on the right indicates that the chance of observing large $\alpha$ in the $\beta$MWM data is somewhat too high. This behavior is improved significantly by adding point mass multipliers in the fine scales.

ponent and thus displays valuable information about the occurrence of rare events such as bursts (small $\alpha$). Figure 5.6 reveals a rich multifractal spectrum. In contrast, fBm has a trivial spectrum consisting only of one point indicating that it has the same “burstiness” everywhere [106].

**Synthetic data**

Having established the LRD and multifractal characteristics of the BC-pAug89 trace, we will next model these properties using the $\beta$MWM. To train the $\beta$MWM, we use the approach outlined in Section 5.3.6. We choose the number of wavelet scales $n = 16$ to synthesize data sets of $2^{16}$ points. This allows us to collect multiple realizations of the wavelet coefficients and root scaling coefficient, and thus form reliable mean and variance estimates. For the root scaling coefficient, we use the Gaussian assumption discussed in Section 5.3.4.

With trained $\beta$MWM in hand, we synthesize 15 length-$2^{16}$ sub-traces and concatenate
them to form a trace of approximately length-$10^6$, the size of the real data set. We now apply
the same battery of tests to this trace as we applied to the actual Bellcore $BC$-$pAug89$ data.

Figure 5.3 compares the $BC$-$pAug89$ with synthetic MWM and dfGn data, at different
aggregation levels. Both models match the mean, variance, and covariance decay of the
real data. Evident from the figure are the large number of (unacceptable) negative values of
dfGn, caused by the real data having a high standard-deviation-to-mean ratio. The MWM
data much more closely matches the characteristics of the real data. Moreover, a length-$2^{18}$
MWM synthesis required just eight seconds of workstation run time, in contrast to eighteen
hours for a Levinson dfGn synthesis.

Figure 5.3(b) shows that the synthesized data captures much of the gross structure of the
Bellcore data at different aggregation levels, including the one-sided marginal density. In
addition, the variance-time plots of Figure 5.4 depict an excellent match of the covariance
structure.\footnote{We remind the reader that the variance-time plot must be interpreted with care due to the non-stationarity
of the wavelet-synthesized data.}

We next measure the multifractal properties of the synthetic trace. From the linearity
of the log-log plots in Figure 5.7(a), we see that the synthetic trace exhibits a multifract-
tal scaling, except for $q$ strongly negative and $j$ large. In converting these plots into the
multifractal spectrum of Figure 5.6, we see that spectrum of the synthesized data closely
matches the $BC$-$pAug89$ spectrum for $\alpha$ near one. The close match in the upper left part,
which corresponds to $q$ values ($=slopes$ of tangents to the spectrum) between 0 and 2, indi-
cates that the $\beta$MWM matches these low ($q$th) order moments very well. The divergence of
the spectra on the right indicates that the chance of observing large $\alpha$ in the $\beta$MWM data
is somewhat too high. Since large $\alpha$ correspond to fast decay, this means that the $\beta$MWM
trace has values that are too small. In fact, the minimum value of the wavelet-synthesized
trace is on the order of $10^{-12}$, whereas the minimum of $BC$-$pAug89$ is on the order of $10^{-5}$.
This is due to the fact that, unlike the coarser-scale $\beta$ multipliers $A_{j,k}$, the fine-scale $\beta$ mul-
tipliers have pdfs with significant mass near $\pm 1$. Clearly, from (5.7) we see that this results
in small values for the synthesized process $C^{(L)}[k]$. This may be indicative of different phenomena in the fine scales of the real data as compared to the coarse scales.

Using $\beta$ distributions in the coarse scales and point mass distributions in the fine scales, we can largely correct this problem, synthesizing data with a minimum value of $10^{-6}$ while preserving the other features of the $\beta$MWM (see Figure 5.6). We choose the point mass parameters (see Section 5.3.3) to match both the wavelet energy decay and the scaling of the negative first moment of the real data in (5.14). We do not claim that the point mass multipliers are realistic — using point mass multipliers at all scales results in syntheses that look somewhat artificial. Here, we simply illustrate the fact that we can choose the multiplier distributions to better match higher-order or lower-order moments of the data.

Figure 5.7: (a) Log-log moment scaling and (b) incremental scaling for the $\beta$MWM synthesized data. (See Figure 5.5 for more description.) The synthetic data exhibits a linear multifractal scaling, with the exception of strongly negative $q$'s and large $j$.

5.4.2 Modeling packets-per-time

The second data set (LBL-TCP-3) contains two hours' worth of wide-area TCP traffic between the Lawrence Berkeley Laboratory and the rest of the world [79] (see Figure 5.8). This data contains the following information about each packet: the time-stamp, (renumbered) source host, (renumbered) destination host, source TCP port, destination TCP port,
and number of data bytes. In our experiments we use only the time-stamp and data bytes information. We form a bytes-per-time trace by counting the number of bytes of packets that arrive in consecutive time intervals of 6 ms and use the first $2^{20}$ data points in our simulation experiments. This trace has a sample mean of 257.5 bytes/(unit time) and sample standard deviation of 562.6 bytes/(unit time).

To fit the WIG and MWM models to the LBL-TCP-3 data, we use the procedure outlined in Section 5.3.6. For the WIG we fit the wavelet variances directly through the $\sigma_j^2$'s, rather than through the $p_j$'s of the MWM model.

We estimate these wavelet variances at the 15 finest scales, because at coarser scales there are not a sufficient number of coefficients to obtain good variance estimates. As a result, we synthesize data traces of maximum length $2^{15}$ data points. For both the MWM and WIG, we model the coarsest-scale scaling coefficient $U_{0,0}$ as a Gaussian random variable with mean and variance equal to the sample mean and variance of the scaling coefficients of the real data at this scale. With trained models in hand, we now generate synthetic data traces.

From Figure 5.8, we see that visually the synthetic MWM looks very similar to the real trace, while the WIG traces have a considerable number of negative points, a result of the low mean and high standard deviation of the real data trace. We also compare the marginals of MWM and WIG traces to that of the LBL-TCP-3 trace at three different aggregation levels. From Figure 5.9 observe that the MWM marginals are similar to that of the real data trace, while the Gaussian WIG marginals differ significantly, even at coarser scales. Note that despite the improved fit the MWM requires no more parameters than the WIG model.

We next compare the variance-time plots of the real data, the MWM traces, and the WIG traces in Figure 5.10(a). The variance-time plot estimates were obtained by averaging the empirical variance-time plots of 32 independent realizations of the models. We observe that, as expected, both the MWM and WIG models do a good job of matching the covariance structure of the real data.

We plot the multifractal spectra of the LBL-TCP-3 data and the synthetic MWM trace in
Figure 5.8: Bytes-per-time arrival process at different aggregation levels for (a) wide-area TCP traffic at the Lawrence Berkeley Laboratory (trace LBL-TCP-3) [79], (b) one realization of the state-of-the-art wavelet-domain independent Gaussian (WIG) model [59], and (c) one realization of the multifractal wavelet model (MWM) synthesis. The top, middle and bottom plots correspond to bytes arriving in intervals of 6 ms, 12 ms and 24 ms respectively. The top and middle plots correspond to the second half of the middle and bottom plots, respectively, as indicated by the vertical dotted lines. The MWM traces closely resemble the real data closely, while the WIG traces (with their large number of negative values) do not.

Figure 5.10(b) (calculations for the negative moments of the WIG data become numerically unstable and hence the spectra for the WIG is not included). Just as for the Bellcore BC-pAug89 data, we observe that the multifractal spectra match extremely well except for large values of $\alpha$ and less well for small $\alpha$. This corresponds to a close match of the
Figure 5.9: Histograms of the bytes-per-times process at different aggregation levels for (a) wide-area TCP traffic at the Lawrence Berkeley Laboratory (trace LBL-TCP-3) [79], (b) one realization of the WIG model, and (c) one realization of the MWM synthesis. The top, middle and bottom plots correspond to bytes arriving in intervals of 6 ms, 12 ms and 24 ms respectively. Note the large probability mass over negative values for the WIG model.

scaling of higher-order moments, with somewhat less accuracy in the scaling of the negative moments.

5.4.3 Queuing behavior

For networking purposes, the final test of the accuracy of the $\beta$MWM model is its queuing behavior. For the Bellcore BC-pAug89 interarrival times, it has been demonstrated in [89]
that the averaged queue performance (measured in terms of the probability that the queue exceeds a certain size) of several MWM traces matches that of the actual data quite closely. Furthermore, when the original trace is subdivided into several parts and fed through the queue, the variability of the MWM of equal size is similar to that of the actual data.

For the Berkeley *LBL-TCP-3* data, it has been observed that the MWM traces closely match the queuing behavior of the real data traces, while the WIG traces do not [86]. Recall the WIG and MWM have identical mean and covariance structures — they differ in that the MWM is non-Gaussian. Hence, this result implies that the non-Gaussianity in LRD network traffic can have a major impact on network performance and validates the MWM approach. Moreover, [86] has derived analytical queuing approximations that track the measured tail queue probabilities of the MWM quite closely. These analytical results hold promise for applying the MWM to more than just synthesis. For instance, in to predict how an actual network will perform, we could measure the MWM parameters of real traffic in real time and substitute these into an analytical queuing formula.
5.5 Discussion

The multifractal wavelet model (MWM) combines the power of multifractals with the efficiency of the wavelet transform in a flexible framework natural for characterizing and synthesizing positive LRD data. As our numerical experiments have shown, the MWM is particularly suited to the analysis and synthesis of network traffic loads. In addition, the MWM could find application in areas as diverse as financial time-series characterization, geophysics (using 2-d and 3-d wavelets and quadrees and octrees), and texture modeling. We also believe the MWM could spur the germination of cross-disciplinary insights between signal processing and the theory of multifractals.

Several extensions to the model hold promise:

1. A parametric characterization of the wavelet-domain energy decay (rather than the current empirical variance measurements) would yield a more parsimonious and robust model.

2. The choice of $\beta$-distributed wavelet multipliers $A_{j,k}$ is not essential. As illustrated by our preliminary work with point mass distributions, we can use distributions with more parameters to match both wavelet energy decay and the scaling coefficient moments.

3. Insights from the multifractal theory can be leveraged into more general multiplicative constructions. For instance, cascades can be used with randomized shifts or non-dyadic structures can be used to obtain stationary processes.

We find that the MWM serves as a simple, efficient, and effective model for positive-valued LRD processes, hence fulfilling our second primary objective outlined in Chapter 1.2. In line with our third primary objective, the MWM holds promise as a tool for the modeling, synthesis, and analysis that are needed to characterize and unravel the effects of non-Gaussianity and LRD in network traffic.

Overall, the MWM's main drawback is that it cannot model the behavior of LRD processes exactly. In the next chapter, we derive a fast algorithm for exactly and efficiently
synthesizing a wide class of Gaussian and non-Gaussian LRD processes.
Chapter 6

FFT-Based Synthesis Of $1/f$ And Long-Range Dependent Data

Although the MWM provides a simple, fast, and powerful model for positive LRD processes, it is inexact. First, the MWM only approximates the covariance behavior of a process. Second the MWM signal is actually nonstationary in covariance, and thus approximates the covariance behavior of a stationary signal only in a time-averaged sense.

In some instances we desire an exact fit to a model, because it allows us to be confident that the observed behavior and properties of a process are authentic, not the result of potential artifacts in the modeling or synthesis procedure. Exactness is particularly relevant for fBm and fGn, since they are widely used in a number of scientific engineering disciplines. Despite their popularity, fBm and fGn are difficult to synthesize accurately, requiring order $O(N^2)$ for an exact synthesis. This requirement can be overburdening for the oftentimes huge data sizes required for accurate simulations involving LRD properties.

However, the Fast Fourier Transform (FFT) allows us to exactly synthesize a wide range of $1/f$-type processes with a much lower complexity $O(N \log N)$ than currently-used Levinson $O(N^2)$ or Cholesky factorization algorithms $O(N^3)$. In this chapter, we prove that our FFT synthesis is exact not only for 1-D fBm and fGn, but for a host of Gaussian LRD processes.

Leveraging the core FFT synthesis algorithm, we develop a framework for modeling and synthesizing general Gaussian and nonGaussian LRD processes. For instance, we derive the covariance structure necessary to obtain self-similar scaling over two different sets of scaling ranges, i.e., processes that exhibit one type of fBm-like over short time intervals and entirely different type of fBm-like scaling over long time intervals. This
behavior has been discovered in network data [36] and may have important consequences for network management [71]. In addition, this model may provide a useful link to the processes that occur in natural science, where it is claimed that for most "fractal" data sets a power-law scaling behavior occurs over a limited scaling range [5]. Finally, by applying a nonlinear transformation to Gaussian LRD processes, we can also extend our approach to handle nonGaussian LRD processes such as ESS noise.

6.1 Basic Approach

The FFT synthesis approach is based on the concept of embedding a Toeplitz covariance matrix into a circulant covariance matrix. This has been studied previously in several contexts [23, 27, 46, 72]. The work in [23, 72] provide procedures for embedding an $M \times M$ positive-definite covariance matrix $R$ into a circulant matrix of size $N \times N$, $N \geq 2M - 2$. Unfortunately, $N$ can become extremely large depending on the condition number of the matrix $R$.

The work in [27] develops several conditions under which the embedding is exact for $N = 2M - 2$ and uses this embedding to derive an algorithm for general FFT-based synthesis of Gaussian processes. The work in [46], independently derived, specifically applies the approach to synthesize two-dimensional fBm, but does not rigorously analyze the exactness of the synthesis. Indeed, in two dimensions it turns out the algorithm is not always exact for synthesizing fBm. We will follow along the lines of [27], but will specialize to the case of $1/f$ and LRD data, demonstrating that the FFT-based synthesis is exact for many Gaussian and nonGaussian LRD processes of interest.

6.1.1 FFT properties and application

The FFT (2.11) has many magical and wondrous properties. Among them is the eigenvalue decomposition it provides for a circulant matrix $C$ [6]. If we denote the top row as
\[
[C(0) \ c(1) \ \ldots \ c(N - 1)], \text{ we can write the entire circulant matrix as}
\]
\[
C = \begin{bmatrix}
c(0) & c(1) & c(2) & \cdots & c(N - 1) \\
c(N - 1) & c(0) & c(1) & \cdots & c(N - 2) \\
c(N - 2) & c(N - 1) & c(0) & \cdots & c(N - 3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
c(1) & c(2) & c(3) & \cdots & c(0)
\end{bmatrix}. \tag{6.1}
\]

Any circulant matrix \( C \) can be written as \( C = \frac{1}{N} F \Lambda F^H \), with \( \Lambda \) a diagonal matrix. The vector \( \lambda \equiv [\Lambda(1, 1) \ \Lambda(2, 2) \ \ldots \ \Lambda(N, N)] \) is the FFT of the vector [\( c(0) \ c(1) \ \ldots \ c(N - 1) \)]. This is exactly the decomposition we desire for the covariance matrix \( R_X \) of a vector \( X \sim N(0, R_X) \) drawn from a stationary process with underlying covariance function \( r_X(n) \). If \( R_X \) were equal to some circulant matrix \( C \), we could synthesize \( X \) via \( X = F \left( \frac{1}{N} \Lambda \right)^{1/2} W \), with \( W \sim N(0, I) \). We can easily check that \( X \) has the correct covariance in this case:

\[
\mathbb{E}[XX^H] = \frac{1}{N} F \Lambda^{1/2} \mathbb{E}[WW^H] \Lambda^{1/2} F^H = C. \tag{6.2}
\]

For a real second-order-stationary process, the covariance matrix \( R_X \) for a length-\( M \) vector \( X \) is formed from the covariance function \( r_X(n) \) according to

\[
R_X = \begin{bmatrix}
\vdots & \vdots & \vdots & \ddots & \vdots \\
r_X[M - 1] & r_X[M - 2] & r_X[M - 3] & \cdots & r_X[0]
\end{bmatrix}. \tag{6.3}
\]

This matrix is Toeplitz, but not necessarily circulant. In general, the FFT will diagonalize such a matrix only as \( M \to \infty \).

However, we can embed \( R_X \) into a \( 2M - 2 \times 2M - 2 \) circulant matrix \( R_Z \) with first row equal to

\[
r_Z \equiv [r_X[0] \ r_X[1] \ \ldots \ r_X[M - 2] \ r_X[M - 1] \ r_X[M - 2] \ \ldots \ r_X[2] \ r_X[1]^T]. \tag{6.4}
\]
The entry in the $i$th row and $j$th column of $R_Z$ is given by

$$R_Z[i,j] = \begin{cases} 
    r_X[i - j], & i, j \leq M \text{ or } i, j > M \\
    r_X[2M - i - j], & j \leq M, i > M \text{ or } i \leq M, j > M
\end{cases} \quad (6.5)$$

Note that $R_Z[i,j] = R_X[i,j]$ for $i,j \leq M$. Thus, our plan will be to synthesize a vector $Z \sim N(0,R_Z)$ of length-$2M - 2$, and then form $X$ as the first $M$ elements of $Z$.

In accordance with (6.2) we could construct $Z = F(\frac{1}{2M - 2}\Lambda)^{1/2}U$, with $U \sim N(0,I)$. Unfortunately, since $F$ is a complex matrix, $Z$ is in general complex. However, if we add a imaginary component $iV$, $V \sim N(0,I)$, $E[UV^T] = 0$, to the spectrum (corresponding to a random phase), we can generate two real sequences with the desired covariance: $X_1 = \text{Re}\left(F(\frac{1}{2M - 2}\Lambda)^{1/2}(U + iV)\right)$ and $X_2 = \text{Im}\left(F(\frac{1}{2M - 2}\Lambda)^{1/2}(U + iV)\right)$.

6.1.2 Algorithm

Putting these ideas together, we present the algorithm to generate two real, length-$M$, mutually independent Gaussian random vectors $X_1, X_2$ with mean $\mu_X$ and covariance function $r_X[n]$:

**FFT Algorithm for Synthesizing LRD Processes**

1. Form the covariance function of the embedding process,

$$R_Z[n] = \begin{cases} 
    r_X[n], & 0 < n \leq M - 1 \\
    r_X[2N - n + 1], & M - 1 < n \leq 2M - 2
\end{cases} \quad .$$

2. Calculate the FFT power spectrum of the embedding process, $\lambda = FR_Z$.

3. Generate a length-$2M - 2$ realization of complex white Gaussian noise $W = U + iV$, $U, V \sim N(0,I)$, $E[UV^T] = 0$.

4. Set $\tilde{\lambda}[k] = \max(0,\lambda[k])$, $k = 0, 1, \ldots, 2M - 2$.

5. Compute $Z$ as the IFFT of the sequence $\left(\frac{\tilde{\lambda}[k]}{2M - 2}\right)^{1/2}W[k]$, $0 \leq k \leq 2M - 2$. In matrix form, $Z = F^H(\frac{1}{2M - 2}\Lambda)^{1/2}W$. 
6. Set \( X_1[n] = \text{Re}(Z[n]) + \mu_X \) and \( X_2[n] = \text{Im}(Z[n]) + \mu_X, \quad n = 0, 1, \ldots, M - 1 \).

7. To generate additional realizations, return to 3.

Note that an embedding of size \( N = 2M - 1 \) can be applied as well [27]. For software that utilizes power-of-two FFT algorithms, an efficient synthesis will occur if \( M = 2^k + 1, \ k \in \mathbb{N} \), leading to an \( N = 2M - 2 \) that is a power of two. For other lengths, efficient algorithms such as the prime factor algorithm also exist [11].

We have glossed over the fact that the embedding process \( Z \) may not even exist. In this case, \( X \) cannot be synthesized exactly through this approach. In general, \( R_Z \) may not be positive semi-definite and, hence, we can obtain \( \lambda_j \)'s that are negative, implying negative power at certain frequencies. In this case, the best we can hope to do is to set the negative \( \lambda_j \)'s to zero, as shown in Step 4. From Parseval's theorem, the mean-squared deviation of the obtained covariance function \( \tilde{R}_Z[n] \) from the desired covariance function will be

\[
\sum_{n=0}^{2M-3} (R_Z[n] - \tilde{R}_Z[n])^2 = \frac{1}{2M-2} \sum_{j \text{ s.t. } \lambda_j < 0} \lambda_j^2. \tag{6.6}
\]

### 6.1.3 Sufficient conditions for exactness

For many cases of practical interest, we can guarantee that \( R_Z \) is positive semi-definite [27], and hence that the algorithm is exact. We will develop several sufficient conditions based on the properties of the covariance function \( r_X[n] \).

**Theorem 1** If the entries of the covariance vector \( r_X = [r_X[0], r_X[1], \ldots, r_X[M - 1]]^T \) satisfy the condition

\[
r_X[0] \geq 2 \left| \sum_{n=1}^{M-2} r_X[n] \right| + |r_X[M - 1]| \tag{6.7}
\]

then the matrix \( R_Z \) in (6.5) is positive semi-definite, and the FFT-based synthesis of Section 6.1.2 is exact.
Proof: Rewrite (2.11) with \( N = 2M - 2 \) applied to \( r = r_Z \) from (6.4) to obtain

\[
S_Z[k] = r_X[0] + (-1)^k r_X[M - 1] + 2 \sum_{n=1}^{M-2} r_X[n] \cos \left( \frac{2\pi nk}{2M-2} \right), \quad k = 0, 1, \ldots, 2M-2.
\]

(6.8)

The theorem follows from the triangle inequality and \(|\cos(x)| \leq 1, \forall x \in \mathbb{R}|.

The next theorem will provide the primary theoretical justification behind the algorithm. But first, let the first and second differences be denoted \( \Delta r[n] \equiv r[n + 1] - r[n] \) and \( \Delta^2 r[n] \equiv \Delta r(n + 1) - \Delta r[n] \), respectively. A sequence \( \{r[n], \ n = 0, 1, \ldots, N\} \) is convex if its second difference satisfies

\[
\Delta^2 r[n] \geq 0, \quad n = 0, 1, \ldots, N - 2.
\]

(6.9)

**Theorem 2** If the entries of the covariance vector \( r_X = [r_X[0], \ r_X[1], \ldots, \ r_X[M - 1]] \) are convex, decreasing, and nonnegative, then \( R_Z \) in (6.5) is positive semi-definite, and the FFT-based synthesis of Section 6.1.2 is exact.

Proof: See [27]. Under the above conditions, the FFT series in (6.8) can be written as a sum of nonnegative Fejér kernels with nonnegative boundary terms.

We will apply both Theorem 1 and Theorem 2 to demonstrate the validity of our \( 1/f \) synthesis. It might appear that the conditions in Theorem 2 are overly conservative and can be broadened, since the proof involves writing the FFT spectrum in (6.8) as the sum of strictly nonnegative terms. In practice we have observed this to be the case — covariance functions deviating slightly from the conditions of Theorem 2 lead to positive-definite \( R_Z \)'s as well. However, the complicated nature of the Fejér kernels makes it difficult to significantly improve the bound. Fortunately, the conditions of Theorem 2 are broad enough to include many LRD covariance functions of practical interest.

### 6.1.4 Proof of exactness for \( 1/f \) Processes

We will now establish the exactness of the FFT-based method for synthesizing dfGn for \( 0 < H < 1 \). By cumulatively summing the dfGn, we obtain exact sampled fBm for
0 < H < 1 as well. Although dfGn for 0 < H ≤ 1/2 is not LRD and hence not of primary interest, we include it for completeness.

Recall the covariance function of a dfGn process \( X_H \) (2.15). We apply Theorem 1 to show for 0 < H < 1/2, that (2.15) satisfies the desired conditions for any \( M > 1 \). We will apply Theorem 2 to show for 1/2 < H < 1 that (2.15) satisfies the desired conditions for 1/2 < H < 1 and any \( M > 1 \).

**Case I: 0 < H ≤ 1/2**

We verify the conditions of Theorem 1 through the following set of inequalities:

\[
\begin{align*}
    r_{X_H}[0] = \sigma_X^2 \quad &\geq \quad \sigma_X^2 \left( 1 - \frac{1}{2} \left( M^{2H} - (M-2)^{2H} \right) \right) \\
     &\geq \quad \sigma_X^2 \left( 1 + (M-2)^{2H} - (M-1)^{2H} \right) \\
     &\quad - \frac{\sigma_X^2}{2} \left( (M-2)^{2H} - 2(M-1)^{2H} + M^{2H} \right). \quad (6.10)
\end{align*}
\]

By substituting in (2.15), cancelling common terms, and observing that \( H < 1/2 \), we obtain

\[
2 \left| \sum_{n=1}^{M-2} r_{X_H}[n] \right| \quad = \quad 2 \left| \sigma_X^2 \left( -1 - (M-2)^{2H} + (M-1)^{2H} \right) \right| \\
\quad = \quad \sigma_X^2 \left( 1 + (M-2)^{2H} - (M-1)^{2H} \right). \quad (6.12)
\]

Substituting (6.12) into (6.11) leads to

\[
r_{X_H}[0] \geq 2 \left| \sum_{n=1}^{M-2} r_{X_H}[n] \right| + |r_{X_H}[M-1]|. \quad (6.13)
\]

Hence, from Theorem 1, the synthesis is exact for \( M > 1 \) and 0 < H ≤ 1/2.

**Case II: 1/2 < H < 1**

We leave the proof that \( r_{X_H} \) satisfies the conditions of Theorem 2 to Appendix B. However, it is easily verified for \( n \) large. Recall that the covariance function \( r_{X_H}[n] \) is one-half the central second difference of the structure function \( \sigma_X^2 |n|^{2H} \). For large \( n \), the second difference approximates the second derivative, so the claim that the sequence \( r_{X_H}[n] = \)
\( \sigma^2_X |n|^{2H} \) is positive, decreasing, and convex can be easily verified from the derivatives of \( \sigma^2_X |t|^{2H} \). For the case \( n = 0, 1 \), the second difference differs considerably from the second derivative; this case must be treated separately.

6.2 Exactness for Two Classes of Gaussian LRD Processes

It may be unrealistic to model real-world processes using a strict fGn or fBm process, with an entire covariance structure determined through the choice of the single \( H \) parameter. Although real-world processes may exhibit the same LRD as fGn or fBm, the short-term covariances may act behave differently, and for certain applications the behavior of these short-term covariances is of great importance. For instance, in our canonical networking example, we expect that both LRD and short-term covariances will play a major role in queuing behavior. We investigate two parametric models that overcome this limitation and prove that the FFT synthesis is exact for these two models.

6.2.1 Asymptotic discrete fractional Gaussian noise

The first class, proposed by Kaplan and Kuo [48], results from the following covariance

\[
r_X[n] = \begin{cases} 
\sigma^2_X, & n = 0 \\
\frac{\sigma^2_X}{2} \left[(A - 1)(1 - |p|)|p|^{n-1} + A \left(|n + 1|^{2H} + |n - 1|^{2H} - 2|n|^{2H}\right)\right], & |n| > 1
\end{cases}
\]

(6.14)

with \( A = \frac{2H + p(2 - 2H)}{2H - p(2 - 2H)} \). They call this process asymptotic discrete fractional Gaussian noise (adfGn) and design a wavelet-based algorithm for estimating its parameters from observed data. Just as for standard dfGn, \( 1/2 < H < 1 \) models the long-range dependence of the process. However, with the additional term corresponding to the parameter \( p \), we have more flexibility in modeling the short-range dependence of the process. Notice the covariance corresponds to the sum of the covariance of an AR(1) process with the covariance of fGn.

We verify that (6.14) satisfies the conditions of Theorem 2 by observing that \( |p|^{n-1} \) is convex, decreasing, and nonnegative for \( n > 0, |p| < 1 \) and that the sum of two convex,
decreasing, and nonnegative sequences is also convex, decreasing, and nonnegative. Hence, we can apply the FFT-based method of Section 6.1 to exactly synthesize adfGn processes modeled via (6.14).

6.2.2 Kinked fractional Brownian motion

Other processes of interest exhibit different self-similar scaling over different scaling ranges [36]. To model such processes, consider a process with a structure function of the form:

\[
g_B[n] = \begin{cases} 
\sigma_X^2 |n|^{2H_1}, & 0 \leq n \leq n_1, \\
\gamma |n|^{2H_2}, & n_2 \leq n < \infty,
\end{cases}
\]

with \(1/2 < H_1, H_2 < 1, \gamma > 0, \) and \(n_2 > n_1 > 0\). We call such a process kinked fBm (kfBm) and the corresponding increments process \(X\) kinked discrete fraction Gaussian noise (kdfGn). These are so named, since the variance-time plot exhibits two straight line segments connected with a “kink” (see Figure 6.3(b)). We focus on \(H_1, H_2 > 1/2\), since this corresponds to a process with LRD behavior.\(^1\)

Over short time scales, \(n < n_1\), the process behaves like sampled fBm with parameter \(H_1\). Over longer time scales, \(n > n_2\), the process behaves like fBm with parameter \(H_2\). A variance-time plot of such a process has a slope of \(2H_1 - 2\) for \(n < n_1\) and \(2H_2 - 2\) for \(n > n_2\).

Although we define kfBm and kdfGn in terms of the structure function of kfBm, we will actually synthesize these processes by generating the kdfGn increments process. Hence, we must derive the kdfGn covariance function \(r_X[n]\). Observe that \(g_B[n]\) is derived from

---

\(^1\)It is easily shown that the special case of \(H_1 > 1/2\) (LRD power-law scaling behavior over fine scales) combined with \(H_2 = 1/2\) (independence over coarse scales) can be obtained by selecting \(r_X[n]\) to be equal to (2.15) for \(0 \leq n \leq n_1\) and equal to 0 for \(n > n_2\). This model could be useful for processes such as those described in [5].
$r_X[n]$ via the difference equation:

$$g_B[n + 1] = 2g_B[n] - g_B[n - 1] + 2r_X[n], \quad (6.15)$$

with boundary conditions $g_B[0] = 0$ and $g_B[1] = \sigma_X^2$. We will invert this equation to solve for $r_X[n]$ in terms of $g_B[n]$.

**Sampled kFbm without a transition region**

Let us construct a kdfGn covariance function to model the kFbm scaling behavior. The simplest example results from $n_2 = n_1 + 1$. That is, we piece-together the kdfGn covariance function as follows:

$$r_X[n] = \begin{cases} \frac{\sigma_X^2}{2} (|n + 1|^{2H_1} - 2|n|^{2H_1} + |n - 1|^{2H_1}), & 0 \leq n \leq n_1, \\ \frac{\gamma}{2} (|n + 1|^{2H_2} - 2|n|^{2H_2} + |n - 1|^{2H_2}), & n_1 + 1 \leq n < \infty. \end{cases}$$

From (6.15) and (6.2.2), the structure function of kFbm is then given by

$$g_B[n] = \begin{cases} \sigma_X^2|n_1|^{2H_1}, & 0 \leq n \leq n_1 \\ \gamma|n_1 + 1|^{2H_2} + \alpha, & n = n + 1 \\ \gamma|n_1 + 2|^{2H_2} + \beta, & n = n + 2 \\ \gamma|n_2 + 2|^{2H_2} + \beta(n - n + 1) - \alpha(n - n + 2), & n > n + 2 \end{cases} \quad (6.16)$$

with

$$\alpha = 2\sigma_X^2|n_1|^{2H_1} - \sigma_X^2|n_1 - 1|^{2H_1} - 2\gamma|n_1|^{2H_2} + \gamma|n_1 - 1|^{2H_2} \quad (6.17)$$

$$\beta = 3\sigma_X^2|n_1|^{2H_1} - 2\sigma_X^2|n_1 - 1|^{2H_1} + 2\gamma|n_1 - 1|^{2H_2} - 3|n_1|^{2H_2}. \quad (6.18)$$

Since the error terms $\alpha$ and $\beta$ are in general non-zero, with this approach it is impossible to exactly achieve the desired scaling behavior for $n > n_2$. Moreover, from (6.16), in general the deviation from strict self-similarity grows linearly with time for $n > n_2$. However, if $\gamma$ is chosen via

$$\gamma = \sigma_X^2 \left( \frac{|n_1|^{2H_1} - |n_1 - 1|^{2H_1}}{|n_1|^{2H_2} - |n_1 - 1|^{2H_2}} \right) \quad (6.19)$$
then it is easy to show that $\alpha = \beta$ and that

$$
g_B[n] = \begin{cases} 
\sigma^2_X [n]^{2H_1}, & 0 \leq n \leq n_1 \\
\gamma |n|^{2H_2} + \alpha, & n > n_1
\end{cases}
$$

(6.20)

Thus, for $\gamma$ as in (6.19), the process $B$ will exhibit the desired self-similar scaling asymptotically as $n \to \infty$. In practice, since $g_B[n]$ increases at a super-linear rate for $H_2 > 1/2$, the deviation for $n > n_1$ is fairly minor.

With the $\gamma$ of (6.19), the covariance function for $n > n_1$ is

$$
r_X[n] = \begin{cases} 
\frac{\sigma^2_X}{2} (|n - 1|^{2H_1} - 2|n|^{2H_1} + |n + 1|^{2H_1}), & 0 \leq n \leq n_1, \\
\frac{|n_1|^{2H_2} - |n_1 - 1|^{2H_2}}{|n_1|^{2H_2} - |n_1 - 1|^{2H_2}} \frac{\sigma^2_X}{2} (|n + 1|^{2H_2} - 2|n|^{2H_2} + |n - 1|^{2H_2}), & n_1 + 1 \leq n < \infty.
\end{cases}
$$

Note that the covariance function jumps substantially (up if $H_1 < H_2$, down if $H_2 > H_1$) in the transition from $n_1$ to $n_1 + 1$ (see Figure 6.1).

![Figure 6.1: Covariance function for kdfGn with $H_1 = 0.75$, $H_2 = 0.9$, $n_1 = 32$, and $n_2 = 33$ (top), and $H_1 = 0.75$, $H_2 = 0.6$, $n_1 = 32$, and $n_2 = 33$ (bottom). Each covariance function shares the same values for $n \leq n_1 = 32$. At $n = n_1$, the kdfGn covariance functions exhibit a discontinuity — a jump up for $H_2 = 0.9$ and a drop for $H_2 = 0.6$. In some cases, this behavior may not correspond to a valid covariance function; however, it can be avoided by using a transition region (see Section 6.2.2).](image)

**Sampled kfBm with linear transition**

Aside from the fact that such a jump appears somewhat artificial in an otherwise smooth covariance function, the resulting covariance function may not even be valid (positive semi-definite). Even if valid, this function will not satisfy Theorem 2 and, hence, we may not be
able to exactly synthesize processes with this covariance exactly via our FFT approach. For this reason, we will utilize a transition region that ensures the covariance remains smooth and convex, yet still provides the desired scaling behavior.

In the transition region \( n_1 < n < n_2 \), we will choose the covariance function to be a linear function \( an + b \), leading to the overall expression:

\[
    r_X[n] = \begin{cases} 
        \frac{\sigma^2_X}{2} \left[ |n + 1|^{2H_1} + |n - 1|^{2H_1} - 2|n|^{2H_1} \right], & 0 \leq |n| \leq n_1 \\
        an + b, & n_1 < |n| < n_2 \\
        \gamma \left[ |n + 1|^{2H_2} + |n - 1|^{2H_2} - 2|n|^{2H_2} \right], & n_2 \leq |n| \leq \infty
    \end{cases}
\]  

(6.21)

We use a linear transition for two reasons. First, it is analytically tractable and always leads to a valid solution. Second, from the following heuristic argument we can demonstrate that a linear function provides an efficient (short) transition.

For \( H_1 > H_2 \), the value of the covariance function going into the transition is too large and must be reduced. Thus, in the transition we wish to decrease the covariance function as quickly as possible, yet maintain convexity and monotonicity of the overall covariance function. In this case, a linear transition function with slope \( a = \Delta r_X[n_1 - 1] \) leads to the shortest transition for a fixed decrease in the covariance function (see Figure 6.2).

For \( H_2 > H_1 \), the value of the covariance function going into the transition is too small. Hence, in the transition we wish the covariance function to decay as slowly as possible, yet still maintain convexity and monotonicity. In this case, a linear transition function with slope \( a = \Delta r_X[n_2] \) leads to the smallest decrease in the covariance function for a given transition length (see Figure 6.2).

Having settled on a linear transition, we will take the following approach. Given the values \( \sigma^2_X, H_1, H_2, n_1, \) and \( n_2 \), we will solve for \( a, b, \) and \( \gamma \) to approximate the desired scaling behavior (6.2.2). As in (6.20), we will have to tolerate a slight error term in the structure function for \( n > n_2 \). For \( n_2 \) chosen large enough, we will demonstrate that the covariance function remains convex, decreasing, positive, and, if viewed as a continuous-time function, smooth.
Figure 6.2: Covariance function for kdfGn with $H_1 = 0.75$, $H_2 = 0.9$, $n_1 = 32$, and $n_2 = 248$ (top), and for kdfGn with $H_1 = 0.75$, $H_2 = 0.6$, $n_1 = 32$, and $n_2 = 82$ (bottom). In each case, $n_2$ is chosen to be the smallest integer for which the covariance function remains convex. For $n \leq n_1$, the covariance function corresponds to dfGn with $H = H_1$. Starting from $n = n_1$, the kdfGn covariance function is linear (dotted line) until $n_2$ is reached, after which the covariance function corresponds to dfGn with $H = H_2$.

For the covariance function of (6.21), the structure function can be found through (6.15) to be

$$g_B[n] = \begin{cases} 
\sigma_X^2 |n|^{2H_1}, & 0 \leq n \leq n_1, \\
2\sigma_X^2 n_1^{2H_1} - \sigma_X^2 (n_1 - 1)^{2H_1} + 2(an_1 + b), & n = n_1 + 1 \\
h[n], & n_1 + 1 < n < n_2 \\
\gamma n_2^{2H_2} + \alpha, & n = n_2 \\
\gamma (n_2 + 1)^{2H_2} + \beta, & n = n_2 + 1 \\
\gamma (n + 1)^{2H_2} + \beta(n - n_2) - \alpha(n - n_2 - 1), & n > n_2 + 1
\end{cases}, \quad (6.22)$$

with

$$h[n] = \frac{a}{3} n^3 + bn^2 + \left(a(-n_1^2 + n_1 - \frac{1}{3}) + b(-2n_1 + 1) + \sigma_X^2 n_1^{2H_1} - \sigma_X^2 (n_1 - 1)^{2H_1}\right) n + a\left(\frac{2n_1^3}{3} - n_1^2 + \frac{n_1}{3}\right) + b\left(n_1^2 - n_1\right) + \sigma_X^2 \left((-n_1 + 1)n_1^{2H_1} + n_1(n_1 - 1)^{2H_1}\right),$$

(6.23)

$$\alpha = -\gamma (n_2)^{2H_2} + h(n_2),$$

(6.24)

and

$$\beta = 2h(n_2) - h(n_2 - 1) + \gamma (n_2 - 1)^{2H_2} - 2\gamma (n_2)^{2H_2}.$$  
(6.25)
As in (6.20), we set $\alpha = \beta$ so that the structure function (6.22) will be in error by only a fixed constant for $n \geq n_2$. This leads to the equation
\[
\gamma \left( n_2^{2H_2} - (n_2 - 1)^{2H_2} \right) = h(n_2) - h(n_2 - 1).
\] (6.26)

Although our covariance function is in discrete-time, we will choose $b$ and $\gamma$ to ensure the "continuity" of the covariance function at $n_1$ and $n_2$, respectively via
\[
b = r_X(n_1) - an_1
\] (6.27)
\[
\gamma = \frac{2(an_2 + b)}{(n_2 - 1)^{2H_2} - 2n_2^{2H_2} + (n_2 + 1)^{2H_2}}.
\] (6.28)

Substituting $h$ into (6.26) and then plugging in (6.27) for $b$ and (6.28) for $\gamma$ leads to:
\[
a = \frac{r_X(n_1) (2n_2 - \eta - 2n_1) + \sigma_X^2 n_1^{2H_1} - \sigma_X^2 (n_1 - 1)^{2H_1}}{(n_2 - n_1) (\eta + 1) - (n_2 - n_1)^2},
\] (6.29)
\[
\text{with } \eta = \frac{2n_2^{2H_2} - (n_2 - 1)^{2H_2}}{(n_2 - 1)^{2H_2} - 2n_2^{2H_2} + (n_2 + 1)^{2H_2}}.
\] (6.30)

Having solved for the desired parameters, we will now verify that the resulting covariance function is in accordance with Theorem 2, which ensures that FFT-based synthesis of such a process is exact. For a fixed $n_2$, a sufficient condition for the overall covariance function to be convex, decreasing, and positive is that $\gamma > 0$ and
\[
\Delta r_X[n_1 - 1] < a < \Delta r_X[n_2] < 0.
\] (6.31)

We will demonstrate that such an $n_2$ exists.

We will find it useful to approximate differences of a discrete function with the derivatives of the identical function viewed as a continuous function of time, for instance, $\Delta r_X[n_2] \approx r'_X(n_2)$. As $n_2 \to \infty$, these approximations become exact.

We start by substituting derivatives for the first and second differences in the $\eta$ term of (6.29) and finding the limit of (6.29) as $n_2 \to \infty$. This leads to the result
\[
\lim_{n_2 \to \infty} a = \frac{2^{2H_2} - 2r_X[n_1]}{3 - 2H_2} \to 0,
\] (6.32)

Since $a \to 0$ and $\Delta r_X[n_1 - 1] < 0$, for $n_2$ large enough we find that $a > \Delta r_X[n_1 - 1]$. 

Substituting (6.27) and (6.32) into (6.28), we observe that

$$
\lim_{n_2 \to \infty} \gamma = \frac{2H_2 - 1}{3 - 2H_2 (n_2 - 1)^{2H_2} - 2n_2^{2H_2} + (n_2 + 1)^{2H_2}} \frac{r_X[n_1]}{r_X[n_2]}.
$$

(6.33)

Hence for $n_2$ large enough, $\gamma > 0$.

Finally, by substituting (6.33) into (6.21), approximating the second difference via the second derivative, and taking a derivative with respect to $n$, we obtain

$$
\lim_{n_2 \to \infty} \Delta r_X(n_2) = \lim_{n_2 \to \infty} r_X'(n_2) = \frac{(2H_2 - 1)(2H_2 - 2)r_X[n_1]}{(3 - 2H_2)n_2} \approx \frac{2H_2 - 1}{2} a.
$$

(6.34)

Since $a < 0$ and $\frac{2H_2 - 1}{2} < 1$, from (6.34) we see that for $n_2$ large enough $a < \Delta r_X[n_2]$.

Hence, the conditions $\gamma > 0$ and (6.31) are satisfied, guaranteeing that an $n_2$ exists for which the covariance function is convex, decreasing, positive, and consistent with the desired scaling behavior of (6.2.2).

### 6.2.3 Gaussian LRD synthesis example

In Figure 6.3, we provide an example to illustrate our approach. Using the FFT method of Section 6.1.2, we synthesized length-131073 sampled fBms for $H = 0.9$, $\mu_X = 0$, $\sigma^2_X = 1$ and $H = 0.6$, $\mu_X = 0$, $\sigma^2_X = 38.63$ and a sampled kfBm (6.2.2) with $H_1 = 0.9$, $H_2 = 0.6$, $\sigma^2_X = 1$, $\mu_X = 0$, $n_1 = 64$, $n_2 = 340$, and $\gamma = 38.63$. The value $n_2 = 340$ is the minimum $n_2$ for which the covariance function remains convex, positive, and decreasing using a linear transition (as verified through (6.31)). As expected, over short time scales, the kfBm behaves like the fBm with $H = 0.9$ and $\sigma^2_X = 1$. Over larger time scales, the kfBm behaves like the fBm with $H = 0.6$ and $\sigma^2_X = \gamma = 38.63$.

At the bottom of Figure 6.3 are the variance-time plots for the fBm and kfBm traces. Included within these plots are empirical estimates of $H$ corresponding to a linear fit of the log-log plot over the scales indicated. These $H$ estimates agree quite well with the desired theoretical values, as they are within the standard error range of the variance-time plot [101]. To decrease the bias in these plots and $H$ estimators, the variances in the variance-time plot were calculated using the knowledge that the processes were zero-mean. For $H$
Figure 6.3: (a) Sampled fractional Brownian motion (fBm) with \( H = 0.9 \) and \( \sigma_X^2 = 1 \), (b) sampled kinked fractional Brownian motion (kfBm) with \( H_1 = 0.9, H_2 = 0.6, \sigma_X^2 = 1 \), and transition region from \( n_1 = 64 \) to \( n_2 = 340 \), and (c) sampled fBm with \( H = 0.6 \) and \( \sigma_X^2 = 38.63 \). All data sets were synthesized using the FFT method. Top: Plots of the entire length-131073 traces. Over large time scales, the kfBm exhibits a smoothness and persistency similar to fBm with \( H = 0.6 \) (to the right). Middle: Plots zoomed-in to the first 512 samples. Over small time scales, the kfBm exhibits a smoothness and persistency similar to fBm with \( H = 0.9 \) (to the left). Bottom: Variance-time plots. The o's are empirical variance-time plot measurements, and the dotted lines correspond to least-squares linear fits of \( H \). For all traces, the empirical variance-time plots and \( H \) estimates closely match the desired theoretical behavior. The kfBm trace demonstrates a linear scaling behavior corresponding to \( H = 0.9 \) over fine resolution and \( H = 0.6 \) over coarse resolutions, joined by a "kink" in the middle. 6.2.3.
near one, reliable mean estimation is difficult and can lead to inaccuracy in the variance calculations, particularly at coarser resolutions.

6.3 Synthesis of NonGaussian LRD Processes

In many applications, the data networking and the stock market to name two, the LRD processes of interest are strictly positive and can exhibit heavy-tailed, nonGaussian behavior. To model this behavior we apply the following set up. We seek to synthesize a stationary, nonGaussian random process $Y[n]$ with distribution function $F_Y$ and covariance function $r_Y$. As shown in Figure 6.4, we form $Y$ as a memoryless nonlinear function $T$ of a process $X[n]$ with covariance function $r_X$. The question is how to choose $T$ and $r_X$ to obtained the desired $F_Y$ and $r_Y$.

In particular, we will prove that through this approach we can synthesize exactly second-order self-similar (ESS) nonGaussian processes without error. Recall from Section 2.1.2 that a discrete-time process $Y[n]$ is ESS if it has a variance-time plot (hence covariance and power spectrum) identical to that of fGn. Thus ESS processes provide nonGaussian equivalents to the $1/f$ process of fGn.\(^2\)

In this formulation, we do not specify the higher-order covariances of the process. In addition, our approach applies only to finite-variance processes that have a well-defined covariance function, which would exclude, for example, the stable distributions. For a fast, but approximate, synthesis of positive-valued LRD data based on wavelets, see [89].

6.3.1 Creating a nonGaussian process from one Gaussian process

By definition, the random variable $\Phi(X[n])$ is distributed uniformly on $[0, 1]$, and we can apply the inversion principle [26, 41, 55] to synthesize a random process $Y[n]$ with distribution $F_Y$. Let the value $X[n]$ have a Gaussian first-order marginal distribution $N(\mu_X, \sigma_X^2)$.

\(^2\)Note that we slightly prefer the use of ESS over nonGaussian $1/f$ noise, since as noted in Section 2.1, any discrete-time realization of an ESS or fGn process will not be strictly $1/f$. 
and let \( F_Y(y) \) be a continuous distribution function on \( \mathbb{R} \) with inverse \( F_Y^{-1} \) defined by [26]

\[
F_Y^{-1}(u) = \inf \{ y : F(y) = u, \ 0 < u < 1 \}.
\] (6.35)

Then we can synthesize the non-Gaussian process \( Y[n] \) via

\[
Y[n] = T(X[n]) = F_Y^{-1}\left( \Phi\left( \frac{X[n] - \mu_X}{\sigma_X} \right) \right).
\] (6.36)

If \( X \) has covariance function \( r_X[n] \), then the covariance function for \( Y \) can be calculated as follows. Let

\[
\rho_X[n] \equiv \frac{E[X[m]X[m-n]] - \mu_X^2}{\sigma_X^2} = \frac{r_X[n]}{\sigma_X^2}
\] (6.37)

be the normalized covariance function for \( X \). Then \( r_Y[n] \) may be found via

\[
r_Y[n] = E[T(X[m]) T(X[m-n])] - E[T(X[n])^2]
\] (6.38)

with

\[
E[T(X[m]) T(X[m-n])] = \int_{-\infty}^{\infty} \frac{T(x_1)T(x_2)}{2\pi \sigma_X^2 \sqrt{1 - \rho_X^2}} \Upsilon(x_1, x_2, \mu_X, \sigma_X^2, \rho_X^2[n]) \, dx_1 \, dx_2,
\] (6.39)

where

\[
\Upsilon(x_1, x_2, \mu_X, \sigma_X^2, \rho_X^2) \equiv \exp\left( -\frac{(x_1 - \mu_X)^2}{2\sigma_X^2} + \frac{2\rho_X(x_1 - \mu_X)(x_2 - \mu_X)}{\sigma_X^2(1 - \rho_X^2)} \right).
\] (6.40)
Calculating (6.38) leads to the transformation \( r_Y[n] = W(r_X[n]) \), which we invert to obtain the desired "prewarped" covariance function for \( X \) via\(^3\)

\[
 r_X[n] = W^{-1}(r_Y[n]) .
\] (6.41)

### 6.3.2 Creating a nonGaussian process from two Gaussian processes

The inversion technique utilizes the fact that several nonGaussian densities are easily formed by transforming of a uniform density. Unfortunately, the mapping \( \Phi \) needed to obtain a uniform from a Gaussian is difficult to apply and analyze. For many cases, this approach can be cumbersome, since \( \Phi \) requires numerical integration and for certain densities \( F^{-1} \) and \( W \) may be difficult to calculate. Let us look to a new approach that is more amenable to closed-form analysis.

There is an extremely simple mapping between two independent Gaussian and a uniform random variable. For a fixed value of \( n \), if \( X_1[n], X_2[n] \sim N(0, 1) \) and are independent, then the random variable

\[
 Y[n] = \exp \left( -\frac{1}{2} \left( X_1^2[n] + X_2^2[n] \right) \right)
\] (6.42)

is uniformly distributed on \((0, 1)\) [45]. This suggests an alternative synthesis approach for synthesizing a random process \( Y \) with first order distribution \( F_Y \). We set

\[
 Y[n] = F_Y^{-1} \left( \exp \left( -\frac{1}{2} \left( X_1^2[n] + X_2^2[n] \right) \right) \right),
\] (6.43)

with \( X_1 \) and \( X_2 \) two independent zero-mean Gaussian random processes with common covariance function \( r_X[n] \).

As before, to synthesize \( Y \) with a desired covariance structure, we must calculate the

\(^3\)Aside from the covariance function \( r_Y \), \( r_X \) may also depend on the parameters of the marginal density of \( Y \) (its mean for example). For clarity of presentation, we suppress the potential dependency of \( r_X \) on these auxiliary parameters, since they can be absorbed into \( W^{-1} \).
warping

\[ W(r_X[n]) = \mathbb{E} [T(X_1[m], X_2[m]) \cdot T(X_1[m - n], X_2[m - n])] - \mathbb{E}^2 [T(X_1[m], X_2[m])] \]

(6.44)

and its inverse \( r_X[n] = W^{-1}(r_Y[n]) \). This leads to an integral similar to (6.39).

Note that since \( Y \) is a function of the square of Gaussian processes, this approach is only valid for synthesizing positively-correlated processes, i.e., processes such that \( r_Y[n] \geq 0 \).

### 6.3.3 Algorithm for synthesis of nonGaussian LRD processes

Putting all of these steps together, we obtain the algorithm for synthesizing nonGaussian LRD processes:

1. Prewarp the desired LRD covariance function \( r_Y \) via \( r_X = W^{-1}(r_Y) \), with \( W \) obtained from (6.38) if \( Y \) is synthesized from one process \( X_1 \) or (6.44) if \( Y \) is synthesized from two processes \( X_1 \) and \( X_2 \). If necessary, calculate \( \mu_X \).

2. Using the FFT-based synthesis algorithm of Section 6.1, synthesize Gaussian random vectors \( X_1, X_2 \) with mean \( \mu_X \) and covariance function \( r_X \).

3. Transform the data via \( Y = T(X_1) \) from (6.36) or \( Y = T(X_1, X_2) \) from (6.43).

As in the Gaussian case, there is not a guarantee that the we can exactly synthesize the target nonGaussian covariance function \( r_Y \). Indeed, the prewarped Gaussian process may not even exist! In some cases the covariance function \( r_Y \) would be valid if \( Y \) were Gaussian, but when prewarped back to \( r_X = W^{-1}(r_Y) \) leads to an invalid covariance function for \( X \) [55].

Fortunately, for ESS processes with the correlation structure of dfGn, for \( 1/2 < H < 1 \) we can use Theorem 2 demonstrate that in many cases the synthesis is exact for the underlying Gaussian covariance function \( r_X \). We have already established that if \( Y \) has a covariance function corresponding to (2.15) then \( r_Y \) is convex, positive, and decreasing. Prewarping typically preserves monotonicity and positivity, but can destroy the convexity
of the covariance function. In Appendix C, we provide general sufficient conditions for
the nonGaussian ESS synthesis to be valid and demonstrate that ESS synthesis is valid for
the lognormal, uniform, exponential and Pareto densities, densities that we now subject to
closer scrutiny.

6.3.4 Example densities

Lognormal

If $X[n]$ has a first-order pdf $N(\mu_X, \sigma_X^2)$, then $Y[n] = \exp(X[n])$ has a first-order lognormal
pdf [45]

$$f_Y(y) = \sigma_y \sqrt{2\pi} \exp\left[-\frac{1}{2} \left(\log y - \mu_X \right)^2 \right], \quad y > 0$$  \hspace{1cm} (6.45)

and mean and variance

$$\mu_Y = \exp \left( \mu_X + \frac{\sigma_X^2}{2} \right) \hspace{1cm} (6.46)$$

$$\sigma_Y^2 = \exp \left( 2\mu_X + 2\sigma_X^2 \right) - \exp \left( 2\mu_X + \sigma_X^2 \right). \hspace{1cm} (6.47)$$

In addition, if $X[n]$ has covariance $r_X[n]$ then from (6.39) it can be shown that $Y[n]$ has
covariance

$$r_Y[n] = \sigma_Y^2 \frac{\exp \left( r_X[n] \right) - 1}{\exp \left( \sigma_X^2 \right) - 1}. \hspace{1cm} (6.48)$$

The transformations (6.46), (6.47), (6.48) are inverted by

$$\mu_X = \log \left( \frac{\mu_Y^2}{\sqrt{\mu_Y^4 + \sigma_Y^4}} \right) \hspace{1cm} (6.49)$$

$$\sigma_X^2 = \log \left( 1 + \frac{\sigma_Y^2}{\mu_Y^2} \right) \hspace{1cm} (6.50)$$

$$r_X[n] = \log \left( 1 + \frac{r_Y[n]}{\mu_Y^2} \right). \hspace{1cm} (6.51)$$
Uniform

Although a random process whose values are uniformly-distributed on \([0, 1]\) can be obtained simply through the transformation \(Y = \Phi(X)\), the transformation

\[
Y[n] = \exp \left( -\frac{1}{2} \left( X_1^2[n] + X_2^2[n] \right) \right)
\]

provides a construction more amenable to closed-form analysis.

By completing the square and using the fact the Gaussian pdf integrates to one, we can show that

\[
\mathbb{E} \left[ \exp \left( -\frac{1}{2} \left( X[m]^2 + X[m-n]^2 \right) \right) \right] = \frac{1}{\sqrt{4 - r_X^2[n]}},
\]

and hence

\[
r_Y[n] = \frac{r_X^2[n]}{4(4 - r_X^2[n])}.
\]

Inverting this, we obtain

\[
r_X[n] = 4 \sqrt{\frac{r_Y[n]}{1 + 4r_Y[n]}}.
\]

To generate a uniform variable on the interval \((a, b)\), we simply form

\[
Y[n] = a + (b - a) \exp \left( -\frac{1}{2} \left( X_1^2[n] + X_2^2[n] \right) \right)
\]

and obtain \(r_X[n]\) as

\[
r_X[n] = 4 \sqrt{\frac{r_Y[n]}{(b - a)^2 + 4r_Y[n]}}.
\]

Exponential

An exponential random variable \(Y\) with \(f_Y(y) = \lambda e^{-\lambda y}, y > 0\) corresponds to \(F_Y^{-1}(y) = -\frac{1}{\lambda} \log y [26]\). Hence, to synthesize an exponential random process with rate \(\lambda\) we form

\[
Y[n] = \frac{1}{\lambda} \left( \frac{X_1^2[n] + X_2^2[n]}{2} \right)
\]
We observe that

\[ \mathbb{E}[Y[n]Y[m-n]] = \frac{1}{4\lambda^2} \left( 2 + 2\mathbb{E}[X^2[m]X^2[m-n]] \right) \]  \hspace{1cm} (6.59)

and thus

\[ r_Y[n] = \frac{1}{\lambda^2} r_X^2[n], \]  \hspace{1cm} (6.60)

implying that

\[ r_X[n] = \lambda \sqrt{r_Y[n]}. \]  \hspace{1cm} (6.61)

In passing, we note that a Weibull random variable with parameters \((\lambda, a)\) may be formed from the \(a\)-th root of an exponential random variable \([26, 100]\). Hence, a Weibull random process may be obtained via

\[ Y[n] = \left[ \frac{1}{\lambda} \left( \frac{X_1^2[n] + X_2^2[n]}{2} \right) \right]^{1/a}. \]  \hspace{1cm} (6.62)

An expression for the warping of the covariance in terms of the hypergeometric function is provided in \([100]\). However, a derivation for the inversion of this expression appears analytically intractable.

**Pareto**

A Pareto\((a, b)\) random variable with \(f_Y(y) = \frac{ab^a}{y^{a+1}}, \ x \geq b > 0\) and

\[ \mathbb{E}[Y] = \frac{ab}{a-1}, \quad (a > 1), \]  \hspace{1cm} (6.63)

\[ \mathbb{E}[Y^2] = \frac{ab^2}{a-2}, \quad (a > 2), \]  \hspace{1cm} (6.64)

\[ \sigma_Y^2 = \frac{ab^2}{(a-1)(a-2)}, \quad (a > 2), \]  \hspace{1cm} (6.65)

has \(F^{-1}(y) = \frac{b}{y^{1/a}}\). All moments of order \(q \geq a\) do not exist.

To form a Pareto random process, we apply \([26]\)

\[ Y[n] = b \exp \left( \frac{X_1^2[n] + X_2^2[n]}{2a} \right). \]  \hspace{1cm} (6.66)
For \( a > 2 \), we find that

\[
\mathbb{E}[Y[m]Y[m - n]] = \frac{a^2 b^2}{a^2 - 2a + (1 - r_X[n])},
\] (6.67)

which leads to the following expression for \( r_Y[n] \) in terms of \( r_X[n] \)

\[
r_Y[n] = \frac{a^2 b^2 r_X[n]}{(a - 1)^4 - r_X[n](a - 1)^2}.
\] (6.68)

To find \( r_X[n] \) in terms of \( r_Y[n] \), we invert as follows

\[
r_X[n] = \frac{(a - 1)^2 \sqrt{r_Y[n]}}{\sqrt{a^2 b^2 + r_Y[n](a - 1)^2}}.
\] (6.69)

By writing \( r_X[n] \) in terms of \( \rho_Y[n] = r_Y[n] / \sigma_Y^2 \), we obtain the simpler expression

\[
r_X[n] = \frac{(a - 1) \sqrt{\rho_Y[n]}}{\sqrt{a^2 - 2a + \rho_Y[n]}}.
\] (6.70)

Note that care must be taken in the use of such a process for \( 2 < a \leq 4 \), since the fourth moment does not exist. Hence, the process may not be covariance ergodic [76] and may vary significantly from realization to realization.

### 6.3.5 NonGaussian synthesis examples

In Figure 6.5 we provide synthesis examples of Gaussian, uniform, exponential, lognormal, and Pareto ESS traces of length-65537, each synthesized using the FFT method of Section 6.3.3. The parameters of each process are chosen so that \( H = 0.85 \), \( \mu_Y = 1 \), and \( \sigma_Y^2 = 1 \). Although all of these processes have the same theoretical second-order statistics, they have different higher-order moments and are quite distinct visually. Hence, we could expect them to behave quite differently in data network simulations, for example. Moreover, we note that although the averaged auto-covariance function is indistinguishable from the true covariance function, in any one realization the covariance can vary significantly from the true auto-covariance. This deviation is more pronounced in LRD data with \( H \) near one, and is exacerbated by skewed distributions with heavy tails, such as the Pareto.\(^4\) In these

\(^4\)Furthermore, the averaged auto-covariances plotted in Figure 6.5 were obtained assuming a known mean.

With LRD data, for any one realization the empirical mean can vary substantially, and covariance estimates can behave even more erratically.
Figure 6.5: Plots of the ESS process $Y$ for (a) Gaussian, (b) uniform, (c) exponential, (d) log-normal, and (e) Pareto data synthesized via the FFT method of Section 6.3.3. The parameters of each process are chosen so that $\mu_Y = 1$, $\sigma^2_Y = 1$, and $H = 0.85$. At top are the full length-65537 traces. At middle are the first 1000 samples. The processes have the same theoretical second-order statistics, but appear strikingly different. At bottom are the averaged auto-covariance function with 90% confidence intervals computed over 500 trials. Except for the Pareto process, the averaged auto-covariance function is nearly indistinguishable from the true covariance function, but in any one realization the covariance can vary significantly from the true auto-covariance, particularly for $H$ near one and for heavier-tailed densities such as the Pareto.

instances, great care must be taken in interpreting and estimating parameters from a single LRD trace.

In Figure 6.6, we plot the prewarped Gaussian covariance functions required to synthesize the traces in Figure 6.5 and the prewarping functions themselves. These plots demonstrate that the underlying Gaussian process exhibits stronger LRD than the resulting non-Gaussian process.

Although the focus of this chapter is not on the measurement and estimation of LRD in Gaussian and non-Gaussian data, this example illustrates how the tools developed here might be useful for such a study. Our experiments indicate that for highly-skewed nonGaus-
Figure 6.6: (a) Gaussian covariance functions (normalized) required to synthesize the traces in Figure 6.5. In all cases, the underlying Gaussian process $X$ is more highly correlated than the target non-Gaussian process $Y$. (b) The normalized Gaussian covariance values $\rho_X[n] = r_X[n]/\sigma_X^2$ as a function of the normalized non-Gaussian covariance values $\rho_Y[n] = r_Y[n]/\sigma_Y^2$.

Gaussian data with $H$ close to one, on any given realization the sample covariance can deviate significantly from the theoretical covariance. In this case, accurate parameter estimation from a single trace is difficult. The FFT algorithm could be used to test the robustness of different LRD parameter estimation algorithms, particularly those derived under a Gaussian assumption [101].

6.4 Discussion

Although continuous-time $1/f$ processes such as fBm and fGn have a number of fascinating properties, at the end of the day a digital computer-based synthesis of these signals must be performed in discrete time. We have demonstrated that an exact synthesis should be based on the time-domain properties of these processes. Moreover, we have proved that an FFT approach based on embedding a Toeplitz matrix into a circulant matrix is exact and, because of its computational superiority ($O(N \log N)$ versus $O(N^2)$), clearly the best method for exactly synthesizing sampled fBm and dfGn.

The simplicity of the time-domain approach combined with the fast FFT algorithm has allowed us to develop a general framework for modeling and exactly synthesizing Gaussian
and nonGaussian LRD processes. For instance, we have introduced a new Gaussian process called kinked fractional Brownian motion. KfBm exhibits different fBm-like scaling behavior over different resolutions of analysis, and for several applications of interest may be a more accurate model than pure fBm. In the nonGaussian arena, we have demonstrated efficient synthesis techniques for exactly second-order self-similar processes, processes with a covariance structure identical to dfGn, but with a nonGaussian first-order density. These types of processes occurs in areas as diverse as as data networking and finance and can be synthesized exactly for many nonGaussian pdfs of interest.

Although we have covered several topics, a few issues beg to be addressed. Synthesis of nonGaussian LRD processes (such as equivalents to kdfGn and adfGn of Section 6.2) via the FFT method is straightforward from the work developed here. In general it is difficult to guarantee that this synthesis procedure is exact, but we have found that if the Gaussian covariance function satisfies Theorem 2, the covariance will often be valid for FFT-based nonGaussian synthesis as well.

Another area of clear interest, particularly in image processing, is synthesis of multi-dimensional LRD processes. Unfortunately, as shown in theory [27] and practice [46], many of the useful one-dimensional conditions for guaranteeing exactness do not hold in higher dimensions. Moreover, for higher-dimensional spaces, there does not exist a single fGn-like increments process for fBm [46]. Generalizations of this work to multi-dimensional processes are quite difficult.

Lastly, to synthesize more general types of nonGaussian LRD data, a general polynomial warping (such as an expansion in terms of Hermite polynomials) [41, 55] would provide some additional modeling flexibility not available using the nonGaussian pdfs derived here. However, in general such a method requires the use of numerical integration to calculate the Hermite polynomial coefficients.

With algorithms to exactly synthesize both Gaussian and nonGaussian LRD processes, we address secondary goal number of this thesis — providing researchers with simulation tools to help differentiate the effects of LRD and nonGaussianity in their system. Since
the synthesis is exact, they will be confident in the fact that the properties of the traces are authentic and not due to some artifact of the synthesis process.

Besides its potential applicability in networking, this work could be useful for applications such as financial modeling, where market volatility is known to exhibit LRD and nonGaussian behavior [10, 62]. More generally, since the synthesis is exact, it could also be used for benchmark testing of LRD parameter estimation algorithms for a variety of processes, both Gaussian and nonGaussian. Finally, this work provides synthesis and modeling tools that could lead to new insights into the general behavior of $1/f$ and LRD processes (particularly nonGaussian), which are still poorly-understood relative to classical Markov, Poisson, and AR processes.
Chapter 7

Discussion And Future Work

7.1 Contributions of This Thesis

The contributions of this thesis come under three major categories.

Wavelet-domain Hidden Markov Models, which combine probabilistic graphs and mixture densities to model the non-Gaussianity and local dependencies of wavelet coefficients.

1. We designed statistical algorithms to test, train, and apply these models.

2. We demonstrated the utility of these models in signal estimation, synthesis, and classification.

3. We developed the framework of probabilistic graph theory for improving and generalizing these models, such as for different transforms.

4. We examined the notion of contexts to increase modeling flexibility without introducing extra computational complexity.

The Multifractal Wavelet Model, a simple, yet effective, multiplicative model for positive-valued $1/f$ data.

1. We developed a designed a general approach for fitting the model to theoretical $1/f$ processes or to empirical wavelet data.

2. We derived several properties of the model and explored its relation to the rich field of cascades and multifractals.
3. We applied the model to network traffic, demonstrating a good fit to marginals, covariance structure, and queueing behavior. This fit compares well with state-of-the-art fractional Gaussian noise and fractional Brownian motion models.

**FFT-based synthesis of \(1/f\) and LRD noise**, a fast, exact algorithm for generating realizations of Gaussian and nonGaussian LRD processes.

1. We provided a general overview of the properties of \(1/f\) processes and algorithms for computer-based simulations of these processes. We have found such an overview is lacking in the literature and believe that it could clear up several popular misunderstandings concerning \(1/f\) processes.

2. We proved that an FFT synthesis based on embedding a Toeplitz covariance matrix into a circulant correlation is exact for fractional Gaussian noise, fractional Brownian motion, and a host of \(1/f\) and long-range dependent processes. This shows that, when exactness is a must, the FFT-based synthesis is the method of choice due to its simplicity and clear computational superiority.

3. We developed kinked fractional Brownian motion, a novel generalization to fBm that has potential use in network simulations and may be useful for modeling "fractal" scientific data, which often exhibits true fractal scaling behavior over only a limited number of scales.

4. We adapted the FFT algorithm to synthesize nonGaussian \(1/f\) noise (e.g. ESS noise), and proved that such a synthesis is exact for several nonGaussian pdfs of interest.

Much of the value of this work derives from the improvements, applications, and generalizations of the work here that have already been developed by other researchers, particularly members of the Rice DSP group [13, 86, 90]. They have addressed several of the open issues of this thesis, and develop vast improvements to this work. It has been gratifying to have in some small part contributed to the genesis of these ideas.
The developments in [13] have led to powerful new wavelet-domain HMM algorithms for multiscale image segmentation. In addition to its use of HMMs, [13] combines the results of several HMM segmentations at different resolutions, resulting in an overall image segmentation that is robust, yet has detail to very fine resolution. The work of [90] develops a Bayesian 1/f-type image model for the HMT model. This approach is potentially more robust and less computationally intensive than estimating a model via the EM algorithm. This work also generalizes the HMT structure to fit an undecimated or shift-invariant wavelet transform, which for several applications is known to be far superior to the standard DWT. This has led to a novel HMM-based denoising algorithm that rivals the best available algorithms for estimating signals in additive white Gaussian noise.

The research in [86] has sparked exciting new results for applying the MWM model as a synthesis tool for networking, and more recently as an analysis tool. Although the former is potentially useful for network design and simulation, the latter points to the actual use of the MWM in real-time applications. For example, we could use the parameters of the MWM model as a measure of the “state of the network” and apply this information to predict queueing performance. Research along these lines has found interest among industrial affiliates in the networking arena.

The work on exact synthesis of LRD noise developed here fills a narrow, but we feel important, gap in the theory of fBm, fGn, LRD, and 1/f processes. We already have plans to apply it in queueing experiments and believe that the fBm and fGn synthesis algorithms alone could see widespread usage.

7.2 Some Open Issues

Let us take a moment to consider some open problems that we find of interest.

First, how does multifractality relate to DSP? Although first results have been promising [89], it is unclear to what extent the theory of multifractality can contribute to DSP (and vice-versa). For the engineer, part of the problem is the technical and abstruse nature of the multifractal literature. Nevertheless, aside from the work performed at Rice, multifractal
models are already being used in DSP for Bayesian image modeling [103] and could play a major role in the future.

Second, we are curious to see what happens when (or if) statisticians, speech processors, and computer scientists bring the full force of their new algorithms to bear on wavelet-domain Hidden Markov Models. As an example, the hidden Markov models developed here pale in complexity to those in speech processing, where hidden Markov models have been studied and used for years. Much could be learned from this and other fields.

For instance, most researchers agree that the EM algorithm is subpar for many problems of interest, particularly the fitting of multi-dimensional mixture models [85]. At Rice, a new algorithm for mixture model estimation has proven extremely robust to outliers [94]. The approach substitutes the maximum-likelihood criterion of the EM algorithm with a least-squares criterion on the estimated pdf. We are interested in applying this approach to the context-based HMMs or (if possible) directly to the HMT.

Third, we have noticed that multiresolution analyses, particularly wavelets, have been effective for a number of statistical algorithm, but time-series prediction has not been one of them. If linear prediction is used, there is no theoretical advantage in applying a wavelet transform.

Intuitively, however, there may be some merit for a wavelet or multiresolution approach. The idea is this: Suppose we estimate the value of the signal $X$ at time $n$ using the wavelet coefficients with support just preceding $n$ (the fact that these wavelet coefficients might not exactly "line-up" can be corrected using a shift-invariant transform). Thus, we estimate the current data value using fine-scale coefficients whose support covers the recent past and coarse-scale coefficients whose support covers the more distant past. The point is to use as much relevant information as possible, while the same time avoid overfitting the data. By "throwing out" the data that a priori we believe to be irrelevant, such as high-frequency glitches from the distant past, we hope to obtain a predictor that is sophisticated, yet less prone to overfitting.

---

1 Although the authors in [103] did not realize at the time that they were using a multifractal construction!
Fourth, the \textit{lifting} interpretation of the wavelet transform has pointed out to us some redundancy in the modeling process [99]. Lifting allows us to interpret the wavelet transform as a prediction and update process in which we predict the signal at the fine scales from the signal at the coarse scales; the wavelet coefficients serve as the prediction error. The wavelet-domain modeling that we have performed can thus be interpreted as a modeling and prediction of the \textit{prediction error}, which seems somewhat inefficient and counterintuitive if the original prediction is done reasonably.

If we are using the wavelet transform for compression or signal estimation (not for analysis), the question arises as to whether it would be better to \textit{jointly} construct an adaptive wavelet transform and probability model at once. For instance, a large wavelet coefficient (prediction error) at one scale increases the probability of a large coefficient (prediction error) at the next. Thus, a large wavelet coefficient could signal to the transform that a singularity is likely, its prediction is not working, and that it should adapt itself accordingly at the next scale. We believe that this and related ideas hold significant promise for improved transform-based DSP.
Appendix A

Appendix — EM Algorithm For Hidden Markov Trees

Although the EM algorithm is classical with a well-known basic structure, the exact EM steps are problem dependent. In fact, the EM steps for estimating the parameters of tree-structured probability models have been derived only recently [80, 91], with work primarily focusing on trees of discrete-valued random variables. Following [91], we will develop an EM algorithm for HMTs generalized to handle continuous-valued wavelet coefficients and specialized to the tree structure provided by the wavelet transform. The major change from [91] is that for the leaves of the tree we replace maximum-likelihood (ML) estimation of probability mass function (pmf) values with ML estimation of Gaussian mixture means and variances. In addition, we will demonstrate the effect of tying on the algorithm.

In applying the EM algorithm to HMTs, our task is to fit an \( M \)-state HMT model, parametrized by \( \theta = \left\{ p_{S_i}(m), \epsilon_{i,\rho(i)}^{mn}, \mu_{i,n}, \sigma_{i,m}^2 \mid i = 1, \ldots, P; n, m = 1, \ldots, M \right\} \), to \( K > 1 \) trees of observed wavelet coefficients, with \( P \) the number of wavelet coefficients in each tree. We omit modeling the single scaling coefficient associated with each tree; as mentioned in Section 3.4.2, extensions to handle the scaling coefficients are straightforward.

We obtain the \( K \) trees either by wavelet-transforming \( K \) signal observations, each into a single tree, or by wavelet-transforming one signal observation into \( K \) different wavelet trees, as shown in Figure 3.1(b). In the latter case, we actually tie across trees, modeling different trees using the same set of parameters (see Section 3.5.3 for details). The EM steps are identical for either case.

---

\(^1\)Except for \( i \)'s corresponding to root states, the pmfs \( p_{S_i}(m) \) are completely specified by the root state pmfs and the transition probabilities \( \epsilon_{i,\rho(i)}^{mn} \).
Recall from Section 3.5.1 that the EM algorithm is iterative, and that for HMTs it converges to a local-maximum of the incomplete log-likelihood $f(w|\theta)$. The iterative structure is as follows:

**EM algorithm for HMT**

**Initialize:**

Select an initial model estimate $\theta^0$.

Set iteration counter $l = 0$.

1. **E step:**

Calculate $p(S|w, \theta^l)$, the joint pmf for the hidden state variables (used in the maximization of $E_S \left[ \ln f(w, S|\theta) \big| w, \theta^l \right]$).

2. **M step:**

Set $\theta^{l+1} = \arg \max_{\theta} E_S \left[ \ln f(w, S|\theta) \big| w, \theta^l \right]$.

3. Set $l = l + 1$. If converged, then stop; else return to E step.

We will suppress the superscript $l$ when it is clear from the context.

Since the wavelets coefficients are conditionally Gaussian given the hidden states, the M step involves simple closed-form updates for the means, variances, and transition probabilities as a function of the wavelet coefficients $w$, marginal state pmfs $p\left(S_i = m \big| w, \theta^l\right)$, and parent-child pmfs $p\left(S_i = m, S_{\rho}(i) = n \big| w, \theta^l\right)$. To perform the M step updates, in the E step we calculate these pmfs. In the E step we can also calculate likelihoods, such as $f(w|\theta)$, that can be used for classification and other tasks.

To keep things clear and simple, we will first develop the E step for a single tree. (The E step is often referred to as the forward-backward algorithm in the HMM literature [84] and as the upward-downward or inward-outward algorithm in the artificial intelligence literature [80,91,97].) We will then develop the E step for multiple trees and the M step.

We will finish by incorporating into the EM steps the notion of tying within trees from Section 3.5.3.
We first focus on processing a single size-$P$ wavelet tree containing observed wavelet coefficients $w = [w_1, w_2, \ldots, w_P]$ having hidden states $S = [S_1, S_2, \ldots, S_P]$ that take on values $m = 1, \ldots, M$. The primary task of the E step is to calculate the hidden state probabilities $p(S_i = m|w, \theta)$ and $p(S_i = m, S_{\rho(i)} = n|w, \theta)$. To obtain these probabilities, we introduce a number of intermediate variables.

A.1 Set-up

We now introduce some notation for trees of observed wavelet coefficients, as in Figure 3.1. Similar in structure to the trees of Figure 3.5, these trees are formed by linking the wavelet coefficients rather than the hidden states. We define $T_i$ to be the subtree of observed wavelet coefficients with root at node $i$, so that the subtree $T_i$ contains coefficient $w_i$ and all of its descendants. Now if $T_j$ is a subtree of $T_i$ (i.e., $W_j$ and all its descendants are members of $T_i$), then we define $T_{i\backslash j}$ to be the set of wavelet coefficients obtained by removing the subtree $T_j$ from $T_i$. Without loss of generality, we order $w$ so that $w_1$ is at the root of the entire tree. Thus, $T_1$ is the entire tree of observed wavelet coefficients (a tree-structured version of the vector $w$). In our probability expressions, we will interchange $T_i$ and $w$ when convenient.

For each subtree $T_i$, we define the conditional likelihoods

$$\beta_i(m) \equiv f(T_i | S_i = m, \theta)$$  \hspace{1cm} (A.1)

$$\beta_{i,\rho(i)}(m) \equiv f(T_i | S_{\rho(i)} = m, \theta)$$ \hspace{1cm} (A.2)

$$\beta_{\rho(i)\backslash i}(m) \equiv f(T_{\rho(i)\backslash i} | S_{\rho(i)} = m, \theta),$$ \hspace{1cm} (A.3)

and the joint probability functions

$$\alpha_i(m) \equiv p(S_i = m, T_{i\backslash i} | \theta),$$ \hspace{1cm} (A.4)

with $S_i$ taking discrete values and the coefficients in $T_{i\backslash i}$ taking continuous values.

Based on the HMT properties from Section 3.4.2, the trees $T_i$ and $T_{i\backslash i}$ are independent given the state variable $S_i$. This fact, along with the chain rule of probability calculus, leads
to the desired state probabilities in terms of the $\alpha$'s and $\beta$'s. First we obtain

$$p(S_i = m, T_i|\theta) = \alpha_i(m) \beta_i(m)$$  \hfill (A.5)$$
and

$$p(S_i = m, S_{\rho(i)} = n, T_i|\theta) = \beta_i(m) \epsilon_{i,\rho(i)}^{mn} \alpha_{\rho(i)}(n) \beta_{\rho(i)\iota}(n).$$  \hfill (A.6)$$

The likelihood of $w$ is then

$$f(w|\theta) = f(T_i|\theta) = \sum_{m=1}^{M} p(S_i = m, T_i|\theta) = \sum_{m=1}^{M} \beta_i(m) \alpha_i(m).$$  \hfill (A.7)$$

Bayes rule applied to equations (A.5)–(A.7) leads to the desired conditional probabilities

$$p(S_i = m|w, \theta) = \frac{\alpha_i(m) \beta_i(m)}{\sum_{n=1}^{M} \alpha_i(n) \beta_i(n)}$$  \hfill (A.8)$$
and

$$p(S_i = m, S_{\rho(i)} = n|w, \theta) = \frac{\beta_i(m) \epsilon_{i,\rho(i)}^{mn} \alpha_{\rho(i)}(n) \beta_{\rho(i)\iota}(n)}{\sum_{n=1}^{M} \alpha_i(n) \beta_i(n)}.$$  \hfill (A.9)$$

A.2 E step for a single wavelet tree (upward-downward algorithm)

All state variables within our HMT model are inter-dependent; in determining probabilities for the state variables, we must propagate state information throughout the tree. The upward-downward algorithm is an efficient method for propagating this information. The Up step calculates the $\beta$'s by transmitting information about the fine-scale wavelet coefficients to the states of the coarse-scale wavelet coefficients; the Down step calculates the $\alpha$'s by propagating information about the coarse-scale wavelet coefficients down to the states of the fine-scale wavelet coefficients. Combining the information from the $\alpha$'s and $\beta$'s via equations (A.8)–(A.9), we obtain conditional pmfs for the state of each wavelet coefficient in the tree.
For our derivation, we will focus on models with mixing components that are Gaussian, with density

$$g(w; \mu, \sigma^2) \equiv \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left( -\frac{(w - \mu)^2}{2\sigma^2} \right).$$ \hfill (A.10)

More general densities can also be treated. Recall that we assign to each node \(i\) in the tree a scale \(j = J(i) \in \{1, \ldots, L\}\), with \(j = 0\) the coarsest scale and \(j = L - 1\) the finest scale. Also recall that \(\rho(i)\) is the parent of node \(i\) and \(\mathsf{ch}(i)\) the set of children to node \(i\).

**Up step**

**Initialize:** For all state variables \(S_i\) at the finest scale \(j = L - 1\), calculate for \(m = 1, \ldots, M\):

$$\beta_i(m) = g(w_i; \mu_{i,m}, \sigma_{i,m}^2).$$ \hfill (A.11)

1. For all state variables \(S_i\) at scale \(j\), compute for \(m = 1, \ldots, M\):

$$\beta_{i,\rho(i)}(m) = \sum_{n=1}^{M} \varepsilon_{i,\rho(i)}^{mn} \beta_i(n),$$ \hfill (A.12)

$$\beta_{\rho(i)}(m) = g(w_{\rho(i)}; \mu_{\rho(i),m}, \sigma_{\rho(i),m}^2) \prod_{i \in \mathsf{ch}(\rho(i))} \beta_{i,\rho(i)}(m),$$ \hfill (A.13)

$$\beta_{\rho(i)\setminus i}(m) = \frac{\beta_{\rho(i)}(m)}{\beta_{i,\rho(i)}(m)}. \hfill (A.14)$$

2. Set \(j = J - 1\) (move up the tree one scale).

3. If \(j = 0\), then stop; else return to step 1.

**Down step**

**Initialize:** For state variable \(S_1\) at the coarsest scale \(j = 0\), set for \(m = 1, \ldots, M\):

$$\alpha_1(m) = p_{S_1}(m).$$ \hfill (A.15)

1. Set \(j = j + 1\) (move down the tree one scale).
2. For all state variables $S_i$ at scale $j$, compute for $m = 1, \ldots, M$:

$$
\alpha_i(m) = \sum_{n=1}^{M} \epsilon_{i, \rho(i)}^{mn} \alpha_{\rho(i)}(n) \beta_{\rho(i)\backslash i}(n).
$$

(A.16)

3. If $j = L - 1$, then stop; else return to step 1.

### A.3 E step for multiple wavelet trees

To handle $K > 1$ wavelet trees, we add a superscript $k$ to denote the tree number. We denote the observed wavelet coefficients as $w = [w^1 w^2 \ldots w^K]$ and the hidden states as $S = [S^1 S^2 \ldots S^K]$. The vectors $w^k = [w^1_k w^2_k \ldots w^K_k]$ and $S^k = [S^1_k S^2_k \ldots S^K_k]$ contain the wavelet coefficients and states of the $k$th tree, respectively.

To implement the E step at iteration $l$ of the EM algorithm, we apply the upward-downward algorithm independently to each of the $K$ wavelet trees. Using the parameter estimates $\theta = \theta^l$, we calculate the probabilities $p \left( S^k_i = m \mid w^k, \theta^l \right)$ and $p \left( S^k_i = m, S^k_{\rho(i)} = n \mid w^k, \theta^l \right)$ for each tree via equations (A.8) and (A.9).

### A.4 M step

Once the probabilities for the hidden states are known, the M step is straightforward. We update the entries of $\theta^{l+1}$ as follows:

$$
p_{S_i}(m) = \frac{1}{K} \sum_{k=1}^{K} p \left( S^k_i = m \mid w^k, \theta^l \right),
$$

(A.17)

$$
\epsilon_{i, \rho(i)}^{mn} = \frac{\sum_{k=1}^{K} p \left( S^k_i = m, S^k_{\rho(i)} = n \mid w^k, \theta^l \right)}{K p_{S_i}(m)},
$$

(A.18)

$$
\mu_{i,m} = \frac{\sum_{k=1}^{K} w^k_i p \left( S^k_i = m \mid w^k, \theta^l \right)}{K p_{S_i}(m)},
$$

(A.19)
\[ \sigma_{i,m}^2 = \frac{\sum_{k=1}^{K} (w_i^k - \mu_{i,m})^2 \ p(S_i^k = m \mid w^k, \theta^l)}{K \ p_S(m)} \]  

(A.20)

The updates for the state probabilities \( p_S(m) \) and \( e_{i,n,i}^{mn} \) are performed by summing the individual state probabilities and then normalizing so that the probabilities sum to one. Just as for the IM model [85] and the hidden Markov chain model [84], updates for the Gaussian mixture means and variances are performed by a weighted averaging of the empirical means and variances, with the weights chosen in proportion to the probabilities of each mixture.

As should be clear from the E and M steps, the per iteration computational complexity of the EM algorithm is linear in the number of observed wavelet coefficients. The overall complexity may involve a large multiplicative constant depending on the number of hidden states used and the number of iterations required to converge. However, as shown throughout this paper, even the simplest two-state HMT model can capture many densities quite well.

### A.5 Tying within trees

The M step changes slightly when tying is performed within trees, such as tying wavelet coefficients and their states within a certain subband or scale. (See Section 3.5.3 for the basic idea behind tying.) With tying, we perform extra statistical averaging over coefficients that are tied together within each tree. For the \( k \)th tree \( w^k \) with wavelet coefficients \( w_i^k \), we write \( i \sim j \) if \( w_i^k \) and \( w_j^k \) (and their states) are tied, which means that they are modeled with the same underlying density parameters. The set \( [i] = \{ j \mid w_j^k \sim w_i^k \} \) is the equivalence class of \( i \), with \(|[i]|\) the number of elements in the class.

For simplicity, we assume that all trees are tied in the same fashion (that is, the coefficients in the trees \( w^1, w^2, \ldots, w^K \) are tied in the same manner) according to the collection
of equivalence classes given by the \([i]\)'s. In this scenario, the M step becomes:

\[
p_{S_i}(m) = \frac{1}{K} \sum_{k=1}^{K} \sum_{j \in [i]} \frac{1}{[[j]]} p \left( S_j^k = m \big| \mathbf{w}^k, \theta^l \right), \quad (A.21)
\]

\[
\varepsilon_{i,i',p(i)}^{mn} = \frac{1}{K p_{S_{p(i)}}(m)} \sum_{k=1}^{K} \frac{1}{[[i]]} \sum_{j \in [i]} p \left( S_j^k = n, S_{p(j)}^k = m \big| \mathbf{w}^k, \theta^l \right), \quad (A.22)
\]

\[
\mu_{i,m} = \frac{1}{K p_{S_i}(m)} \sum_{k=1}^{K} \frac{1}{[[i]]} \sum_{j \in [i]} w_j^k p \left( S_j^k = m \big| \mathbf{w}^k, \theta^l \right), \quad (A.23)
\]

\[
\sigma_{i,m}^2 = \frac{1}{K p_{S_i}(m)} \sum_{k=1}^{K} \frac{1}{[[i]]} \sum_{j \in [i]} (w_j^k - \mu_{j,m})^2 p \left( S_j^k = m \big| \mathbf{w}^k, \theta^l \right). \quad (A.24)
\]

Although (A.21)-(A.24) appear more computationally intensive than (A.17)-(A.20), the computational complexity remains the same, since the common parameters for each equivalence class \([i]\) are calculated only once.
Appendix B

Exactness Of FFT-Based Synthesis Of DfGn For

\[ 1/2 < H < 1 \]

To verify that our FFT-based synthesis synthesizes dfGn exactly, we must demonstrate the fGn covariance function \( r_{X_H}[n] \) (2.15) satisfies the conditions of Theorem 2. That is, we must show that it is convex, decreasing, and non-negative. Hence, we need to show for any integer \( n > 0 \): \( r_{X_H}[n] \geq 0 \), \( \Delta r_{X_H}[n] \leq 0 \), and \( \Delta^2 r_{X_H}[n] \geq 0 \). We will split this into two cases.

B.1 Case I: \( n \geq 1 \)

It will be convenient to replace the discrete-time function \( r_{X_H}[n] \) with its continuous-time equivalent \( r_{X_H}(\tau) \), since convexity, monotonicity, and nonnegativity for the continuous function \( r_{X_H}(\tau), \tau \in [n_1, n_2], n_1, n_2 \in \mathbb{Z} \) immediately lead to the identical properties in the sequence \( r_{X_H}[n], n = n_1, n_1 + 1, \ldots, n_2 \).

For example, if \( \frac{d^2 r_{X_H}(\tau)}{d\tau^2} \) is non-negative on \( (n_1, n_2) \), then for \( \tau_1 > \tau_2 \), we observe that

\[
\Delta^2 r_{X_H}[n] = (r_{X_H}[n] - r_{X_H}[n+1]) - (r_{X_H}[n+1] - r_{X_H}[n+2]) \quad \text{ (B.1)}
\]

\[
= - \int_n^{n+1} \frac{d r_{X_H}(\tau)}{d\tau} d\tau + \int_{n+1}^{n+2} \frac{d r_{X_H}(\tau)}{d\tau} d\tau \quad \text{ (B.2)}
\]

\[
= \int_n^{n+1} \left[ \frac{d r_{X_H}(\tau+1)}{d\tau} - \frac{d r_{X_H}(\tau)}{d\tau} \right] d\tau \geq 0. \quad \text{ (B.3)}
\]

Hence, it becomes clear that

\[
\frac{d^2 r_{X_H}(\tau)}{d\tau^2} \geq 0, \forall \tau \in (n_1, n_2) \implies \Delta^2 r_{X_H}[n] \geq 0, n_1 \leq n \leq n_2 - 2, \quad \text{ (B.4)}
\]
which from (6.9) implies that the discrete sequence is convex over the interval \([n_1, n_2]\).

In this quest, we will find the following expansion useful for qualitative analysis of \(r_{X_H}(\tau)\) for \(\tau > 1\):

\[
(\tau \pm 1)^c = \tau^c + \sum_{k=1}^{\infty} \frac{(\pm 1)^k \tau^{c-k}}{k!} \prod_{j=0}^{k-1} (c-j)
\]

\[
= \tau^c \pm cr^{c-1} + c(c-1)\frac{\tau^{c-2}}{2!} \pm \cdots
\]  

(B.5) 

(B.6)

By substituting (B.5) into (2.15) and combining terms, we find that for \(\tau > 1\) the series expansion for \(r_{X_H}\) and its first derivatives are thus

\[
r_{X_H}(\tau) = \sigma_X^2 \sum_{k=1}^{\infty} \frac{\tau^{2H-2k}}{(2k)!} \prod_{j=0}^{2k-1} (2H-j),
\]

(B.7)

\[
r'_{X_H}(\tau) = \sigma_X^2 \sum_{k=1}^{\infty} \frac{\tau^{2H-2k-1}}{(2k)!} \prod_{j=0}^{2k} (2H-j),
\]

(B.8)

\[
r''_{X_H}(\tau) = \sigma_X^2 \sum_{k=1}^{\infty} \frac{\tau^{2H-2k-2}}{(2k)!} \prod_{j=0}^{2k+1} (2H-j),
\]

(B.9)

For \(H > \frac{1}{2}\), each term in (B.7) and (B.9) is positive, the product of an even number of negative numbers, while each term in (B.8) is negative, the product of an odd number of negative numbers. For \(0 < H < 1/2\), the opposite statements are true. Thus, it is easily verified that

\[
r_{X_H}(\tau) > \sigma_X^2 H(2H-1)\tau^{2H-2} > 0, \ 1/2 < H < 1, \ \tau > 1,
\]

(B.10)

\[
r'_{X_H}(\tau) < \sigma_X^2 H(2H-1)(2H-2)\tau^{2H-3} < 0, \ 1/2 < H < 1, \ \tau > 1,
\]

(B.11)

\[
r''_{X_H}(\tau) > \sigma_X^2 H(2H-1)(2H-2)(2H-3)\tau^{2H-4} > 0, \ 1/2 < H < 1, \ \tau > 1.
\]

(B.12)

We have thus established that the continuous-time function \(r_{X_H}(\tau)\) is convex, positive, and decreasing for \(\tau > 1\). This fact and the continuity of \(r_{X_H}(\tau)\) at \(\tau = 1\) imply that the discrete-time function \(r_{X_H}[n]\) is convex, nonnegative, and decreasing for \(n \geq 1\).
B.2 Case II: \( n = 0 \)

A continuous-time analysis in this case is more difficult, since the underlying continuous function \( r_{X_n}(\tau) \) is not convex, with a second-derivative undefined for \( \tau = 0, 1 \).

Instead we can look at the first and second differences directly to verify the function is decreasing with positive second difference. At \( n = 0 \), we have

\[
\Delta r_{X_n}[0] = r[0] - r[1] = 2 - 2^{2H-1} \geq 0 \tag{B.13}
\]

\[
\Delta^2 r_{X_n}[0] = \frac{1}{2} \left( 7 - 4 \cdot 2^2 + 3^2 \right) \geq 0 \tag{B.14}
\]

To see the inequality in (B.14), note that

\[
\frac{d(\Delta^2 r_{X_n}[0])}{dH} = 2^{2H} \left( -4 \log 2 + \left( \frac{3}{2} \right)^{2H} \log 3 \right) \leq \frac{2^{2H}}{2} \left( -4 \log 2 + \frac{9}{4} \log 3 \right) \leq 0
\]

(B.15)

with \( \Delta^2 r_{X_n}[0] = 0 \) for \( H = 1 \). Hence, \( \Delta^2 r_{X_n}[0] \) must be positive for \( H \in (1/2, 1) \).
Appendix C

Exactness Of FFT-Based Synthesis Of NonGaussian ESS Processes For \( 1/2 < H < 1 \)

We will verify that the prewarped Gaussian covariance corresponding to synthesis of non-Gaussian \( 1/f \) noise is convex, monotonically decreasing, and positive for several types of nonGaussian pdf functions. Just as in the Gaussian case, we must split the test for convexity conditions into two cases: the case \( n \geq 2 \) and the case \( n = 0, 1 \). Note that we look at the case \( n \geq 2 \) instead of \( n \geq 1 \), since the continuous-time bounds we will derive are not useful for \( n = 1 \).

C.1 Case I: \( n \geq 2 \)

C.1.1 General bound

Consider data with a covariance function \( r_X = W^{-1}(r_Y) \), where \( W^{-1} \) is a twice-differentiable, monotonically increasing map such that \( W^{-1}[1] = \sigma_X^2 \). As in the earlier appendix, we will treat \( r_X \) as a function of a continuous-time lag \( \tau \in \mathbb{R} \), using the fact that if \( r_X(\tau) \) is convex for \( \tau \geq 2 \), then \( \Delta^2 r_X[n] > 0 \) for \( n \geq 2 \).

By the chain and product rules of the calculus, we have

\[
\frac{d^2 r_X(\tau)}{d\tau^2} = \frac{d^2 W^{-1}(r_Y)}{d\tau^2}(\tau) \cdot \left( \frac{dr_Y(\tau)}{d\tau} \right)^2 + \frac{dW^{-1}(r_Y)}{dr_Y}(\tau) \cdot \frac{d^2 r_Y(\tau)}{d\tau^2}.
\]

(C.1)

We will now the examine properties of the inverse warping function \( W^{-1} \) that are necessary to satisfy that the convexity constraint \( \frac{d^2 r_X(\tau)}{d\tau^2} \geq 0 \).

\[
\frac{d^2 W^{-1}(r_Y)}{dr_Y^2}(\tau) \geq \frac{d^2 r_Y(\tau)}{d\tau^2} \left( \frac{dr_Y(\tau)}{d\tau} \right)^2
\]

(C.2)
This bound holds for general LRD processes, not just for ESS. We will use bounds in terms of the series expansions of the covariance function of ESS and its first two derivatives, as derived earlier in Appendix B.

In the case of a ESS process $Y$, from (B.12) we observe that for $\tau > 1$

$$\frac{d^2 r_Y(\tau)}{d\tau^2} \geq \sigma_Y^2 H(2H - 1)(2H - 2)(2H - 3)\tau^{2H - 4}. \quad (C.3)$$

Next, we find from (B.8) that for $\tau > 1$

$$\frac{dr_Y(\tau)}{d\tau} = \sigma_Y^2 H(2H - 1)(2H - 2)\tau^{2H - 3} \left(1 + \sum_{k=2}^{\infty} \frac{\tau^{2k-2} \cdot \prod_{j=2}^{k-1} (2H - 1 - j)}{(2k!)}\right) . \quad (C.4)$$

It is easily verified that $\frac{2 \cdot \prod_{j=2}^{k-1} [(2H - 1 - j)]}{(2k!)} < \frac{1}{2}$ for $k \geq 2$. Hence, for $\tau > 1$ we obtain

$$\left|1 + \sum_{k=2}^{\infty} \frac{\tau^{2k-2} \cdot \prod_{j=2}^{k-1} (2H - 1 - j)}{(2k!)}\right| \leq 1 + \frac{1}{2} \sum_{k=2}^{\infty} \tau^{2k-2} \leq \left(1 + \frac{1}{2} \frac{\tau^{-2}}{1 - \tau^{-2}}\right). \quad (C.5)$$

Thus, for $\tau \geq 2$ from (C.4) and (C.5) we obtain the bound

$$\left|\frac{dr_Y(\tau)}{d\tau}\right|^2 \leq \frac{49}{36} \sigma_Y^4 H(2H - 1)(2H - 2)^2 \tau^{-4H - 6} \quad (C.6)$$

yielding

$$\frac{d^2 W^{-1}(r_Y)}{dr_Y^2}(\tau) \geq -\frac{36}{49} \frac{\sigma_Y^2 H(2H - 1)(2H - 2)}{\tau^{2H - 2}}. \quad (C.7)$$

Since on the interval $\frac{2H - 3}{2H - 2}$ is at least 2 for $H \in [.5, 1]$, we can write

$$\frac{d^2 W^{-1}(r_Y)}{dr_Y^2}(\tau) \geq -\frac{72}{49} \frac{\sigma_Y^2 H(2H - 1)}{\tau^{2H - 2}} \geq -\frac{72}{49} \frac{1}{r_Y(\tau)}. \quad (C.8)$$

Equation (C.8) is a simple, but somewhat overly restrictive condition to check for the convexity of the covariance function for $n \geq 2$.

For instance, as $\tau$ becomes large, the covariance function $r_Y(\tau)$ is well-approximated as a second-derivative of the structure function $\sigma_Y^2 |\tau|^{2H}$, and it is easy to show that we must
have the asymptotic behavior

\[
\frac{d^2 W^{-1}(r_Y) (\tau)}{d\tau^2} \geq - \frac{2}{r_Y(\tau)}, \quad \tau \to \infty
\]  
(C.9)

We now focus on some specific examples.

### C.1.2 Lognormal density

For the lognormal warping (6.51), we have

\[
\frac{dW^{-1}(r_Y(\tau))}{dr_Y} = (\mu_Y^2 + r_Y(\tau))^{-1}
\]  
(C.10)

\[
\frac{d^2 W^{-1}(r_Y(\tau))}{dr_Y^2} = - (\mu_Y^2 + r_Y(\tau))^{-2}.
\]  
(C.11)

This leads to

\[
\frac{d^2 W^{-1}(r_Y(\tau))}{dr_Y} \leq - (\mu_Y^2 + r_Y(\tau))^{-1} \geq \frac{-1}{r_Y(\tau)},
\]  
(C.12)

which satisfies (C.8).

### C.1.3 Uniform density

For the uniform warping (6.42), we see that

\[
\frac{dW^{-1}(r_Y)(\tau)}{dr_Y} = 2(r_Y(\tau))^{-1/2}(1 + 4r_Y(\tau))^{-3/2}
\]  
(C.13)

\[
\frac{d^2 W^{-1}(r_Y)(\tau)}{dr_Y^2} = -(16r_Y(\tau) + 1)(r_Y(\tau))^{-3/2}(1 + 4r_Y(\tau))^{-5/2}
\]  
(C.14)

and

\[
\frac{d^2 W^{-1}(r_Y(\tau))}{dr_Y^2} \geq -8 - \frac{1}{2r_Y(\tau)}.
\]  
(C.15)
which satisfies (C.8).

C.1.4 Exponential density

For the exponential warping (6.61), we observe that

\[
\frac{dW^{-1}(r_Y)}{dr_Y}(\tau) = \frac{\lambda}{2}(r_Y(\tau))^{-1/2} \quad \text{(C.16)}
\]

\[
\frac{d^2W^{-1}(r_Y)}{dr_Y^2}(\tau) = -\frac{\lambda}{4}(r_Y(\tau))^{-3/2} \quad \text{(C.17)}
\]

and

\[
\frac{dw^{-1}(r_Y)}{dr_Y}(\tau) = \frac{-1}{2r_Y(\tau)}, \quad \text{(C.18)}
\]

which satisfies (C.8).

C.1.5 Pareto density: \( a \geq 3 \)

In this case, we work with the normalized covariance function \( \rho_Y(\tau) = r_Y(\tau)/\sigma_Y^2 \). Since \( \sigma_Y^2 \) just provides a scaling, the bound of (C.8) applies with \( r_Y(\tau) \) replaced by \( \rho_Y(\tau) \). For the Pareto warping (6.70), we obtain

\[
\frac{dW^{-1}(\rho_Y)}{d\rho_Y}(\tau) = \frac{a(a - 1)(a - 2)}{2(\rho_Y(\tau))^{1/2}((a^2 - 2a + \rho_Y(\tau))^{3/2})} \quad \text{(C.19)}
\]

\[
\frac{d^2W^{-1}(\rho_Y)}{d\rho_Y^2}(\tau) = \frac{-a(a - 1)(a - 2)(a^2 - 2a + 4\rho_Y(\tau))}{4(\rho_Y(\tau))^{3/2}(a^2 - 2a + \rho_Y(\tau))^{5/2}} \quad \text{(C.20)}
\]

and

\[
\frac{dw^{-1}(\rho_Y)}{d\rho_Y}(\tau) = -\frac{a^2 - 2a + 4\rho_Y(\tau)}{2\rho_Y(\tau)(a^2 - 2a + \rho_Y(\tau))} \quad \text{(C.21)}
\]

We provide bounds for the case \( a > 4 \), where the fourth moment exists. It can be shown that for \( a > 2 \) that the Pareto satisfies the asymptotic bound of (C.9), but not necessarily that of (C.8).
We bound the numerator from above and denominator from below using the fact that
\[ 0 \leq \rho_Y(\tau) \leq 1 \text{ and } \alpha > 4 \text{ to obtain} \]
\[
\frac{dW^{-1}(\rho_Y)}{d\rho_Y}(\tau) > \frac{7}{6\rho_Y(\tau)},
\]
which satisfies (C.22).

C.2 Case II: \( n = 0, 1 \)

For each of these densities, to guarantee convexity we must also verify:
\[
\Delta^2 r_X[0] = W^{-1}(\rho_Y[0]) - 2W^{-1}(\rho_Y[1]) + W^{-1}(\rho_Y[2]) \geq 0
\]
\[
\Delta^2 r_X[1] = W^{-1}(\rho_Y[1]) - 2W^{-1}(\rho_Y[2]) + W^{-1}(\rho_Y[3]) \geq 0
\]

For these densities, the second differences at 0 and 1 are complicated expressions that are difficult to bound analytically. Therefore, we provide plots that, in conjunction with the smoothness of the underlying functions as function of \( H \), indicate that the second differences \( \Delta^2 r_X[0] \) and \( \Delta^2 r_X[1] \) are non-negative for \( 1/2 \leq H \leq 1 \), with the zero value achieved only at the boundaries \( H = 1/2 \) and \( H = 1 \).

Without loss of generality, we set the parameter \( \lambda = 1 \) for the exponential function and assume the uniform lies in the region \([0, 1]\). (Examination of (6.57) combined with the fact that \( r_Y[n] = \frac{(a-b)^2}{12} \rho_Y[n] \) shows that the choice of region has no effect on the covariance \( r_X[n] \).)

For the Pareto density, we will show that we can set the parameter \( \alpha = 3 \) without loss of generality. The second difference of the underlying Gaussian correlation function is given by
\[
\Delta^2 r_X[n] = \left( \frac{\sqrt{\rho_Y[n]} \sqrt{\rho_Y[n+1]} \sqrt{\rho_Y[n+2]}}{(\sqrt{a^2 - 2a + \rho_Y[n]} \sqrt{a^2 - 2a + \rho_Y[n+1]} \sqrt{a^2 - 2a + \rho_Y[n+2]})} \right) \times (a - 1)
\]
\[
(C.25)
\]
Figure C.1: Demonstration of the convexity of the underlying Gaussian covariance function $r_X[n]$ at $n = 0$ and $n = 1$ for the problem of non-Gaussian ESS noise synthesis. (a) The second differences $\Delta^2 r_X[0]$ and $\Delta^2 r_X[1]$ (or lower bounds to them), which are shown to be non-negative for $H \in [1/2, 1]$ for the exponential, Pareto, and uniform densities. (b) The terms $r_Y[n]r_Y[n+2] - r_Y^2[n+1]$ are shown to be non-negative for $H \in [1/2, 1]$ and $n = 0, 1$. This implies that $\Delta^2 r_X[0] > 0$ and $\Delta^2 r_X[1] > 0$ for the lognormal density.

Since $\rho_Y[n+1] \geq \rho_Y[n+2]$, (C.25) is bounded below as follows:

\[
\Delta^2 r_X[n] \geq (a - 1) \left( \frac{\sqrt{a^2 - 2a + \rho_Y[n+1]} \sqrt{\rho_Y[n] - 2\sqrt{\rho_Y[n+1]} + \sqrt{\rho_Y[n+2]}}}{\sqrt{a^2 - 2a + \rho_Y[n+1]}} \right).
\]

\[
\geq \frac{\sqrt{a^2 - 2a + \rho_Y[n+1]}}{\sqrt{a^2 - 2a + \rho_Y[n]}} \sqrt{\rho_Y[n] - 2\sqrt{\rho_Y[n+1]} + \sqrt{\rho_Y[n+2]}} \quad (C.26)
\]

Since $\rho_Y[n] \geq \rho_Y[n+1]$, (C.26) is monotonically increasing in $a$. Hence, in Figure C.1, we plot (C.26) for $a = 3$, $H \in [1/2, 1]$, $n = 0$, and $n = 1$ in order to show that $\Delta^2 r_X[0]$ and $\Delta^2 r_X[1]$ are non-negative for all $a \geq 3$.

Finally, for the lognormal, we have the extra parameter $\mu_Y^2$ that we wish to remove. Note that the second difference for the lognormal function is given by

\[
\Delta^2 r_X[n] = \log \left( \frac{(\mu_Y^2 + r_Y[n-1])(\mu_Y^2 + r_Y[n+1])}{(\mu_Y^2 + r_Y[n])^2} \right). \quad (C.27)
\]

Since, $r_Y[n-1] + r_Y[n+1] > 2r_Y[n]$ by the convexity of $r_Y[n]$, we see that (C.27) is non-negative if $r_Y[n-1]r_Y[n+1] \geq r_Y^2[n]$. Hence, in Figure C.1(b) we plot $r_Y[n]r_Y[n+1]$
2] \(- r_\gamma^2[n + 1]\) as a function of \(H\) for \(n = 0\) and \(n = 1\) to demonstrate that \(\Delta^2 r_X[0]\) and \(\Delta^2 r_X[1]\) are non-negative.
Bibliography


