August, 1999

LIDS- TH 2456

Research Supported By:

AFSOR Grant F49620-98-1-0349
ONR Grant N00014-91-J-1004
Boston Univ., SubContr. GC123919NGD

Internal Multiscale Autoregressive Processes, Stochastic Realization, and Covariance Extension

Austin B. Frakt

Internal Multiscale Autoregressive Processes, Stochastic Realization, and Covariance Extension

by

Austin B. Frakt

B.S., Applied and Engineering Physics
Cornell University, 1994

S.M., Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 1996

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

August, 1999

Internal Multiscale Autoregressive Processes, Stochastic Realization, and Covariance Extension

by Austin B. Frakt

Submitted to the Department of Electrical Engineering
and Computer Science on July 27, 1999
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The focus of this thesis is on the identification of multiscale autoregressive (MAR) models for stochastic processes from second-order statistical characterizations. The class of MAR processes constitutes a rich and powerful stochastic modeling framework that admits efficient statistical inference algorithms. To harness the utility of MAR processes requires that the phenomena of interest be effectively modeled in the framework. This thesis addresses this challenge and develops MAR model identification theory and algorithms that overcome some of the limitations of previous approaches (e.g., model inconsistency and computational complexity) and that extend the breadth of applicability of the framework. One contribution of this thesis is the resolution of the problem of model inconsistency. This is achieved through a new parameterization of so-called *internal* MAR processes. This new parameterization admits a computationally efficient, scale-recursive approach to model realization. The efficiency of this approach stems from both its scale-recursive structure and from a novel application of the estimation-theoretic concept of *predictive efficiency.* Another contribution of this thesis is to provide a unification of the MAR and wavelet frameworks. This unification leads to wavelet-based stochastic models that are fundamentally different from conventional ones.

A limitation of previous MAR model identification approaches is that they require a complete second-order characterization of the process to be modeled. Relaxing this assumption leads to the problem of covariance extension in which unknown covariance elements are inferred from known ones. This thesis makes two contributions in this area. First, the classical covariance extension algorithm (Levinson's algorithm) is generalized to address a wider range of extension problems. Second, this algorithm is applied to the problem of designing a MAR model from a partially known covariance matrix. The final contribution of this thesis is the development of techniques for incorporating nonlocal variables (e.g., multiresolution measurements) into a MAR model. These techniques are more powerful than those previously developed and lead to computational efficiencies in model realization and statistical inference.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

A foolish consistency is the hobgoblin of little minds,
adored by little statesmen and philosophers and divines.
Ralph Waldo Emerson

Consistency, thou art a jewel.
Anonymous

Only by the strictest of definitions am I the sole author of this thesis. The seeds of nearly every idea in this document were sown in a conversation with a colleague. The strength and perseverance required to nurture the seedlings to fruition were drawn from the comfort and camaraderie of friends and family. Financial support was provided by a National Defense Science and Engineering Graduate Fellowship, my thesis advisor, Alan Willsky, and my parents. It was only through the benefit of my interaction with and the funding provided by others that I was able to accomplish more than I imagined possible in my five years at MIT.

In 1994, I chose to come to MIT in large part because of my desire to work with Alan Willsky and to interact with his students. What impressed me then and now was the breadth and depth of Alan's intellect, his expressive clarity, his energy, and the trust and confidence he has in his students. I was and continue to be equally impressed with the students Alan chooses to advise. By choosing wisely, Alan has created an intellectually and socially vibrant research group that is as close to a family as one is likely to find in academic research. I am deeply appreciative that Alan selected me to join his group and I will cherish the companionship I shared with past and present group members.

In 1997, I surprised myself by choosing a thesis topic that once had frightened me. I had feared the stochastic realization of "tree models" because I didn't understand it. Bill Irving, Mike Daniel, and Paul Fieguth each, patiently, spent many hours explaining their views on the open problems in this area (model consistency, computational complexity, and others). Eventually I realized that, while none understood how to resolve these problems, some excellent starting points were to be found in the final chapters of Bill's and Mike's theses. Vestiges of their ideas can be seen in the work of Chapter 3, Chapter 4, and Chapter 7 of this thesis.

My first research breakthrough can be traced to conversations with Mike Schneider in the summer of 1997. In addition to serving as a trusted friend, public transportation guru, cross-country skiing and hiking consultant, and juggling partner, Mike Schneider, in many ways, has served as a second advisor to me. It was one of his suggestions

that ultimately lead to one of the fundamental ideas in this thesis: predictive efficiency, reviewed in Section 2.4 and applied in Chapter 4 and Chapter 7.

In 1998, Khalid Daoudi tested my understanding of my thesis topic by posing insightful questions. Out of our struggles, his to understand my work and mine to explain it, we formed a partnership that lead to contributions in an area (wavelets) that I would not have made on my own. In many respects, the work of Chapter 5 is more his than mine.

Hanoch Lev-Ari, a member of my thesis committee, introduced me to the problem of covariance extension during his sabbatical at MIT in 1996–1997. The problem intrigued me, and for two years I struggled to understand it in its most general form. In the meantime, Hanoch met with me weekly to listen to my latest technical problem which he often rephrased with stunning simplicity and clarity. In early 1999, I revisited the problem of covariance extension and, together with Hanoch, found a way to address it within the framework of my thesis topic. The fruits of our collaboration are found in Chapter 6.

In addition to those mentioned in previous paragraphs, I have benefited from interaction with many others, in particular: Hamid Krim, perhaps my loudest cheerleader, played an instrumental role in bringing the work of Chapter 5 to publication quickly; my thesis committee members Paul Viola and Michael Jordan have spent many hours listening and reacting to my technical problems and solutions; Jun Zhang, in his sabbatical year, provided valuable suggestions as I struggled to formulate a thesis topic; my officemate Dewey Tucker has always been willing to lend a helping hand (or eye, or ear) and brought errors in an earlier draft of this document to my attention.

Fortunately, there has been much more to my life than proving theorems and developing algorithms. Many of my more recent and most rewarding experiences have been shared with and inspired by Kristina Rodolico. Together, she and I have explored many of the beautiful things in life that cannot, or ought not, be expressed mathematically. Thanks in large part to Kris and her creativity, vision, energy, and love, the last year and a half has been the most joyous of any I've spent on this earth.

Finally, I dedicate this thesis to my first teachers, my parents. From time to time I think about what they have taught me, perhaps unknowingly and implicitly, and how it is reflected in the style with which I have conducted my thesis research and have lead my life. Their lessons have served me well in academics and beyond. In this sense, this thesis, as well as all the achievement it represents, is as much theirs as it is my own.

Contents

List of Figures

List of Tables

Notational Conventions

Symbol	Definition
General Notation	
$\lfloor\cdot\rfloor$	"floor" function, denotes the greatest integer no larger than the argument
$\lceil\cdot\rceil$	"ceiling" function, denotes the least integer no smaller than the argument
$\lvert\cdot\rvert$	for matrix arguments, element-wise absolute value; for set arguments, cardinality
$\lVert\cdot\rVert$	2-norm, unless indicated otherwise
$(\cdot)^T$	matrix or vector transpose
$(\cdot)^{-1}$	matrix inverse
$(\cdot)^{1/2}$	matrix square root, i.e., $A = A^{1/2}A^{T/2}$
$[a : b]$	set of integers between a and b, inclusive
$a : b$	same as $[a : b]$
$[a, b]$	closed interval of the real numbers between a and b
(a, b)	$[a, b] - \{a, b\}$
$(a, b]$	$[a, b] - \{a\}$
$[a, b)$	$[a, b] - \{b\}$
$A(i, j)$	element in the i-th row and j-th column of A
$[A]_{i,j}$	same as $A(i, j)$
$A(\alpha, \beta)$	submatrix of A consisting of elements whose row index is in set α and column index is in set β. A colon (:) in place of α (β) indicates that the row (column) indices are unconstrained
$[A]_{\alpha,\beta}$	same as $A(\alpha, \beta)$
$\delta(\cdot)$	discrete-time Dirac function
ε	mean-square estimation error
$\mathrm{E}[\cdot]$	expected value
$\widehat{\mathrm{E}}[f \mid g]$	linear least-squares estimate of f given g
$\widehat{f}$	linear least-squares estimate of f
f^M	fine-scale process to be modeled
I, I_ℓ	identity matrix; ℓ (when specified) denotes number of rows
$\mathcal{M}_r$	set of matrices with no more than r rows
$\mathcal{M}_r^*$	subset of $\mathcal{M}_r$ consisting of block diagonal matrices

Symbol	Definition
General Notation (continued)	
$O(\cdot)$	computational complexity (operation count) is asymptotically bounded by a constant multiple of the argument
ρ	correlation coefficient
P_g	covariance matrix for g
P_{gf}	cross-covariance of g and f
$P_{f\|g}$	covariance of f conditioned on g
$\mathbb{R}$	real numbers
$\mathbb{Z}$	integers
Graphs and Covariance Extension	
(a, b)	edge between vertices a and b
edge(G)	set of edges of graph G
$G = (V, E)$	denotes a graph defined over vertices V with edges E
G_U	the subgraph of graph G induced by the set of vertices U
$p_{a,b}$	covariance element indexed by (a, b)
P_E	partial covariance matrix in which known elements correspond to edges in E
Q	maximal clique with a and b active elements
ucli(T)	"union of cliques", i.e., for $T = (\mathcal{K}, \mathcal{E})$, denotes the union of the elements in $\mathcal{K}$
U	intersection of the sets A and B which are maximal cliques containing the element a and b, respectively
U_a	union of the singleton set $\{a\}$ and the set U
U_b	union of the singleton set $\{b\}$ and the set U
vert(G)	the set of vertices of graph G
z_D	vector indexed by elements in D
MAR Processes	
$A(\cdot)$	autoregression matrix
$\mathcal{A}_n$	block matrix containing all $A(s)$ for scale n
$C(\cdot)$	measurement equation matrix
$\eta(s)$	indices mapped to node s of an end-point MAR model
$\Phi(\cdot, \cdot\cdot)$	state transition matrix
$Q(\cdot)$	process noise covariance
$\mathcal{Q}_n$	diagonal block matrix containing $Q(s)$ for scale n
$R(\cdot)$	measurement noise covariance
$\Upsilon(\cdot, \cdot\cdot)$	scale transition matrix
$v(\cdot)$	measurement noise
$w(\cdot)$	process noise

Symbol	Definition
Tree-Indexed Processes	
d_s	dimension of vector at node s
d	maximum dimension, i.e., $\max_{s \in \mathcal{S}_0} d_s$
$\imath(s)$	shift of node s
$m(s)$	scale, i.e., tree-distance, from the root node to node s
M	finest scale of a tree
N	total number of fine-scale variables, i.e., $\sum_{s \in \mathcal{T}_0(M)} d_s$
q	number of children associated with a non-leaf node of a tree
r, s, t	variables used to denote nodes of a tree
$s\bar{\gamma}$	parent of node s
$s\alpha_i$	i-th child of node s, $i \in \{1, 2, \ldots, q\}$
$s \wedge t$	common ancestor of s and t with maximal scale
$\mathcal{S}_s$	tree nodes in subtree rooted at s
$\mathcal{S}_s^c$	tree nodes other than those in subtree rooted at s
$\mathcal{T}_s(n)$	tree nodes at scale n descending from s
$\mathcal{T}_s^c(n)$	tree nodes at scale n not descending from s
V_s	local internal matrix at node s
W_s	internal matrix at node s
$x(s)$	(vector) value of a tree-indexed process at node s
x_s^n	sub-process of tree-indexed process indexed by $\mathcal{T}_s(n)$
$x_{s^c}^n$	sub-process of tree-indexed process indexed by $\mathcal{T}_s^c(n)$
x^M	finest-scale sub-process of tree-indexed process
Wavelets	
$a_j(n)$	scaling coefficient at scale j and shift n
$d_j(n)$	detail coefficient at scale j and shift n
h, g	discrete-time analysis filters
$\widetilde{h}$, $\widetilde{g}$	discrete-time synthesis filters
$\phi(\cdot)$, $\widetilde{\phi}(\cdot)$	scaling and dual scaling functions
$\phi_{j,n}$, $\widetilde{\phi}_{j,n}$	scaling and dual scaling basis function at scale j and shift n, i.e., $\phi_{j,n} = \sqrt{2^j}\phi(2^j t - n)$ and similarly for $\widetilde{\phi}_{j,n}$
$\psi(\cdot)$, $\widetilde{\psi}(\cdot)$	wavelet and dual wavelet functions
$\psi_{j,n}$, $\widetilde{\psi}_{j,n}$	wavelet and dual wavelet basis function at scale j and shift n, i.e., $\psi_{j,n} = \sqrt{2^j}\psi(2^j t - n)$ and similarly for $\widetilde{\psi}_{j,n}$
R, $\widetilde{R}$	the supports of h and $\widetilde{h}$

Chapter 1

Introduction

IN the last two decades, multiscale techniques, many of which are based on the wavelet transform, have been widely and successfully applied in signal[1] processing. This is due both to their ability to capture compactly the salient scale-to-scale properties that many signals exhibit and to the efficiency of the algorithms to which they lead. With both of these attractive features in mind, the multiscale autoregressive (MAR) framework was introduced [28–30] to support the development of optimal multiscale statistical signal processing. Recent work [37–40, 96–98, 100] has focussed on systematic approaches to MAR model identification. This thesis extends this recent identification work and develops a consistent and complete realization theory that leads to efficient algorithms.

The utility of the MAR framework has already been established in a wide variety of applications [39, 41, 66–69, 74, 90, 91, 99, 106–108, 111, 114, 128, 162, 165]. The success that the framework enjoys stems from two sources: (1) its ability to model compactly a rich class of phenomena and (2) its ability to address efficiently complications that arise in many signal processing problems. With respect to the former, it has previously been shown that the class of processes that can be effectively modeled within the MAR framework includes one-dimensional Markov processes [127, 129], $1/f$-like phenomena [28, 29, 37, 38, 40, 63, 68, 127, 128], some Markov random fields [127, 129], and others [96, 97, 100].

Fast and flexible signal processing algorithms have been developed for MAR processes. Sample-path generation, linear least-squares estimation [28, 29], and likelihood calculation [127, 130] have computational complexity that scale linearly with problem size (under certain conditions to be discussed in the sequel). Moreover, these algorithms are capable of simultaneously handling a variety of challenging features that are typical of real-world statistical signal processing problems such as:

- large data sets,
- nonstationary processes with correlations at many length- or time-scales,
- irregularly spaced, nonlocal, and multiresolution measurements,

[1] Unless specified otherwise, "signal" refers to signals, images, and higher-dimensional processes.

- the need for error statistics.

While a variety of multiscale frameworks for representing and processing signals have been proposed [12,24–27,29,30,35,36,47,50,62,75,76,83,133,142,178,190–192], only the MAR framework can simultaneously and efficiently address all of the aforementioned challenges.

To harness the power of the MAR framework, of course, requires that the phenomena of interest be effectively modeled in the framework. This thesis develops MAR model identification theory and algorithms that overcome some of the limitations of previous approaches and that extend the breadth of applicability of the framework. In particular, the following topics are addressed:

- model consistency,
- computationally complexity,
- unifying wavelets and MAR processes,
- model realization from an incomplete second-order characterization,
- incorporating nonlocal variables systematically.

These topics, and the way they are addressed, are motivated by the prior work of others, which is reviewed in the following section.

1.1 History of the MAR Framework

Figure 1.1 illustrates the progression of work on MAR processes, algorithms, and applications. While the publications by those named in the figure [12–14,16,28–30,37–41, 43,44,63,66–69,73,74,77–79,96–100,110,111,127–131,140,160,162] do not represent all of the work associated with MAR processes (see, for example, [106–108,114,153,165]), they certainly cover all of the theory and most of the applications. A line linking two authors indicates that the work of the later author builds primarily on that of the earlier author although the research lineage could, in fact, be ordered in other ways.

MAR processes grew out of a research effort begun by Basseville, Benveniste, and Willsky in the late 1980s [13, 14]. This effort was aimed at developing a statistical system theory for multiscale analysis and synthesis of signals. The main result of [13] is to establish that the only suitable parameterization of isotropic processes on dyadic trees is via a generalization of Schur-Levinson parameterization for standard time series. Additional theoretical developments such as lattice structures for whitening such isotropic processes is the topic of [14]. In [16] a deterministic realization theory is developed.

Building on the work in [13, 14], Chou considers a variety of processes indexed by lattice and tree structures. In [12, 28] multiscale stochastic models of and optimal estimation algorithms for processes that are whitened by the wavelet transform are

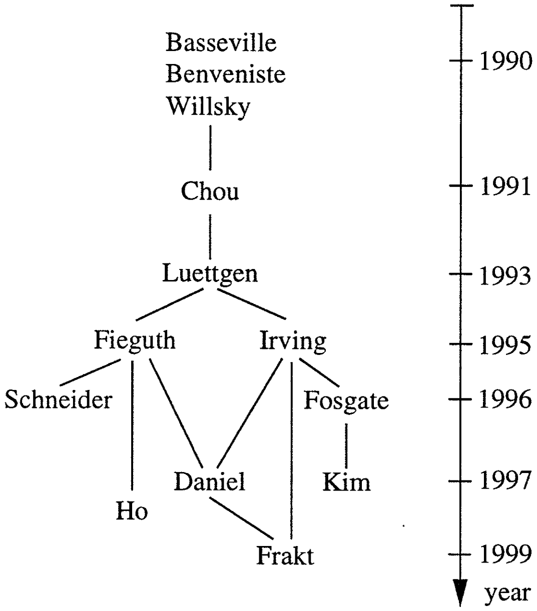


Figure 1.1. MAR research lineage.

developed. Generalizations of the notion of stationarity give rise to several classes of autoregressive processes defined on lattices and trees.

One of these classes is the class of MAR processes which generalize discrete-time state-space processes.[2] Chou developed a computationally efficient linear least-squares estimation algorithm [29] for MAR processes which generalizes the Kalman filter [104, 105] and Rauch-Tung-Striebel smoother [155] for state-space processes. Although not the main focus of his work, Chou provides several one-dimensional examples which hint at the great variety of applications for which the MAR framework is suited. Not surprisingly, $1/f$ processes, which are approximately whitened by the wavelet transform [190–192], are well-approximated by simple MAR models. Chou's work also suggests that one-dimensional Markov processes are also well-approximated.

Once an efficient estimation algorithm had been developed, a natural question to ask was: how rich a class of processes could MAR models capture? This question formed the basis of the work by Luettgen in [127, 129] which represents the first steps toward a MAR stochastic realization theory (although it was not couched in those terms at the time). The work in [127, 129] provides three fundamentally important results. First, it definitively demonstrates that the class of processes effectively modeled by MAR processes is at least as rich as all one-dimensional Markov processes and includes many two-dimensional Markov processes. Second, it shows that to represent a two-

[2] As we will discuss in detail, a MAR process, like a state-space one, consists of a collection of vectors, called *states*, that are related by affine dynamics. The difference is that a MAR process may be indexed by the nodes of *any* tree, while a state-space process is indexed by a monadic tree (i.e., the integers).

dimensional Markov random field exactly with a MAR model requires state dimensions that scale with the linear size of the image leading to statistical inference algorithms that, asymptotically, *do not* scale linearly with problem size. To circumvent this complexity, Luettgen introduces a class of approximate models which brings us to the third significant contribution of his work. What is shown in [127,129] is that the state dimension may be reduced (essentially through subsampling of the wavelet transform) so that the exact desired Markov random field is not captured by the MAR model but, nevertheless, the essential statistical features of the field are retained.

In [127,128] Luettgen provides the first fully developed two-dimensional application of MAR models. Interpreting a quadratic gradient smoothness penalty in a variational formulation of the optical flow problem as a $1/f$ prior and then approximating this $1/f$ prior with a MAR process, Luettgen demonstrates the efficacy of the application of MAR models to the problem of optical flow. In doing so, he suggests (correctly) that MAR models may be successfully applied to a broad range of image processing problems.

Luettgen's other work [127,130,131] is of a more theoretical nature. In [127,131] he develops a statistical characterization of the estimation error associated with linear least-squares estimates of a MAR process. The main result is that the error is also a MAR process. In [127,130] an efficient likelihood calculator is developed and applied to the problem of texture discrimination.

Fieguth primarily builds on the range of applications of MAR models in [63,66,68,69] and provides the first application to an *extremely* large problem—estimating the shape of the north Pacific ocean surface from satellite altimetry measurements. This work is presented in [63,66] and involves the estimation of 100,000 variables from 20,000 measurements. In [63,68], Fieguth applies Luettgen's fast likelihood calculator to the problem of Hurst parameter estimation for fractional Brownian motion. The work of [63,69] addresses the surface reconstruction problem and relies upon an ingenious application of the interpretation of quadratic gradient penalties as fractal priors.

Fieguth's surface reconstruction approach also relies upon the theoretical work of overlapping trees done jointly with Irving and presented in [63,96,97]. The overlapping tree approach offers a way to trade off smoothness of estimates and sample-paths with computational complexity. The motivation for this work is that low dimensional MAR models, while admitting efficient estimation and sample-path generation, can exhibit distracting blockiness in estimates and sample-paths. In some applications this is not a statistically or practically significant issue. However, in others, such as surface reconstruction which requires the calculation of gradients, smoothness *is* a significant issue. An essential feature of Fieguth's and Irving's approach to the smoothness problem is that it *does not* involve low-pass filtering the estimates. Therefore, fine-scale features are *not* smoothed away, just the blocky artifacts.

Several authors have built upon the work of Fieguth. In [160,162] Schneider extends Fieguth's surface reconstruction approach to accommodate the problem of image segmentation. This work is, in essence, a multiscale counterpart to the variational formulation for segmentation proposed by Shah [167]. Ho [67], in collaboration with Fieguth and other authors, extends Fieguth's ocean surface estimation work by simultaneously estimating sea level *and* orbit-induced errors from satellite altimetry data. Ho has also extended the applicability of multiscale models to dynamic problems where the model parameters change over time [90–92].

To a large extent, the work of Irving is motivated by Luettgen's work in [127, 129]. The main focus of Irving's work is the MAR stochastic realization problem [96, 98, 100]. Recall that Luettgen showed that modeling two-dimensional Markov random fields requires prohibitively large state dimensions. Luettgen's approach—subsampling the wavelet transform of Markov random field region boundaries—while effective, is ad hoc and does not suggest a way to model non-Markov textures with reduced-order models.

Irving's approach, on the other hand, provides a general framework for building exact and reduced-order models for *any* process whose second-order statistics are known. Irving's basic idea is to exploit the Markovianity of MAR processes[3] which dictates what information must be kept at each node to model exactly a given process. Irving's approach, based on the concept of canonical correlations [6–8, 49, 54, 94, 96, 100, 183], not only finds this information but prioritizes it. Then, if a reduced-order, approximate realization is desired, the least important information is discarded. Two weaknesses of this approach are addressed in this thesis: (1) it leads to inconsistent models and (2) it is computationally intensive.

Irving's final contribution is a modeling framework for synthetic aperture radar (SAR) imagery [96, 99]. This approach has proved effective for a number of SAR processing problems and has spawned several additional studies. In [96, 99], Irving applies his methodology to automatic target recognition while in [73, 74], Fosgate applies it to segmentation of SAR imagery and the enhancement of anomalies. In turn, Kim builds on Fosgate's work using it as a basis for SAR image compression [110, 111].

Daniel's work [37–41] is of both a theoretical and applied nature and builds on both that of Fieguth and of Irving. Daniel applies the MAR framework to the groundwater hydrology problem of hydraulic conductivity estimation from measurements of conductivity and head in [37, 39] and to the problem of travel time estimation in the advective transport of mass in ground-water aquifers in [37, 41]. One of the primary difficulties in these applications is that they require the fusion of data at different resolutions. While the MAR framework can accommodate such data, doing so requires some care. One of Daniel's contributions in [37, 39] is to show how to augment a MAR model to incorporate multiresolution variables.

In his other work [37,38,40], Daniel focuses on the efficient design of MAR models for self-similar processes. He explores two approaches in an effort to find an approximate MAR representation useful for the generation of fractional Brownian motion sample-paths. One approach involves the synthesis of the wavelet model proposed by Fieguth in [68] with the midpoint displacement model commonly used for fractal Brownian

[3] As we develop in the sequel, a MAR process is a Markov random field on a tree.

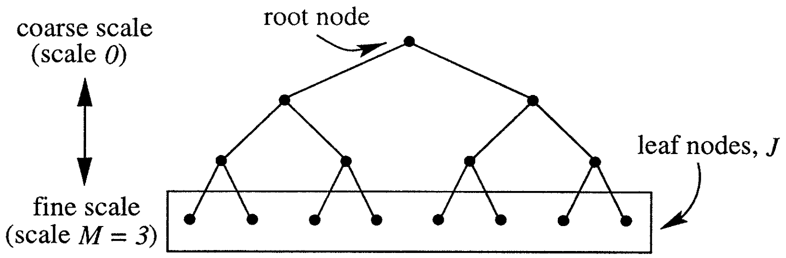


Figure 1.2. A four-scale binary tree.

motion (see [9,129]). The second approach makes clever use of the machinery developed by Irving. In particular, using the stationarity and self-similarity of the increments of fractional Brownian motions, Daniel shows that one can circumvent some of the computational effort required by Irving's canonical correlations approach to the general MAR stochastic realization problem.

While the aforementioned authors have substantially developed the theory, algorithms, and applications of MAR processes, there are a number of important issues that are not addressed by their work. Some of these are raised in the following section and further developed in the body of this thesis. Some of those that remain are posed as challenges to future researchers in Chapter 8.

■ 1.2 Main Problems Addressed

This thesis addresses three main problems which are summarized in this section.

■ 1.2.1 Computationally Efficient Internal MAR Stochastic Realization

A MAR process is a collection of random vectors $\{x(s)\}$, called *states*, each of which is indexed by a node s of a tree. These nodes are organized into scales as indicated in Figure 1.2. MAR states are coupled with affine coarse-scale to fine-scale dynamics that generalize those of a state-space process. One problem addressed in this thesis and which motivates much of the theoretical development of subsequent chapters, is choosing parameters for the dynamics of a MAR process to model the second-order statistics of *any* given fine-scale random signal.

A similar stochastic realization problem arises in the state-space context [151,181]. Although a MAR process is a generalization of a discrete state-space one, the MAR stochastic realization problem *is not* a generalization of the state-space realization problem. To see this, consider a one-dimensional, finite-length signal modeling problem in which case the index set for the signal to be modeled is a subset of the integers (call this subset J). In the state-space setting, the index set for the model is also J. However, in the MAR setting, the index set for the model is a tree whose leaf nodes correspond to J (see Figure 1.2). From a graphical modeling point of view, MAR models, unlike state-space ones, have "hidden nodes" since the statistics to be modeled are provided only for the leaf node states. As shown in Figure 1.2, these hidden nodes are interpreted as residing at coarser scales (with the leaf nodes comprising the finest scale). Building a MAR model requires supplying the "missing" coarse-scale statistical information at the hidden nodes, a step which has no counterpart in state-space modeling.

Although the MAR stochastic realization problem is not a generalization of the state-space one, many of the concepts developed for state-space modeling have useful counterparts in the MAR setting. One of these is the concept of *internality.* An internal state-space process is one for which each state is a linear function of the observed process [123]. The corresponding definition of an internal MAR process is one for which each state $x(s)$ is a linear function of the states indexed by the leaf nodes that descend from node s.

Internal MAR processes, which we discuss in detail in Chapter 3, are important for a variety of reasons. First, internality vastly simplifies model realization because, for an internal process, the statistics for the "hidden" coarse-scale states can be determined from the given finest-scale statistics (i.e., there is no exogenous randomness). Second, as we will discuss, the MAR models developed in [37, 40, 96, 98, 100] are inconsistent precisely because they are not internal (although the intent was to make them so). In contrast, internal models are, by definition, consistent. Third, internal MAR models admit the consistent inclusion of nonlocal linear functions at coarser-scale nodes and, thereby, permit the statistically optimal fusion of multiresolution measurements [39]. The incorporation of nonlocal linear functionals, while aided by the property of internality, is a nontrivial problem in its own right. This thesis presents several techniques for addressing this problem and their development represents important extensions to the theory of internal MAR model realization.

The final important fact about internality is that it plays a role in overcoming the computational burden of previous model building approaches [37, 40, 96, 98, 100]. The computational complexity of these methods stems from two sources. First, they are not scale-recursive and, therefore, do not take advantage of the natural efficiency of tree data structures. Second, they are based on canonical correlations, a burdensome approach involving the inversion and singular value decomposition of large matrices. Consequently, the approach developed in [96, 98, 100] is quartic in problem size while that of [40] is cubic in problem size.[4]

One of the contributions of this thesis is the development of a computationally efficient realization algorithm with complexity quadratic in problem size.[5] The efficiency of this algorithm stems from the fact that it *is* scale-recursive and is *not* based on canonical correlations. With respect to the former, the theoretical basis for the scale-recursive realization algorithm developed in this thesis follows from a thorough analysis of (wide-sense) Markovianity for internal tree-indexed processes. In addition to internality, Markovianity is another important concept in the state-space setting that

[4] The approach in [40] is only applicable to self-similar processes with stationary increments while that of [96, 98, 100], as well as the approach developed in this thesis, is completely general.

[5] As discussed in Chapter 4, an approximation leads to a linear-time algorithm.

generalizes to MAR processes and that plays a central role in the stochastic realization problem. Moreover, and most importantly, it can be shown that, for internal processes, this Markov property has an *equivalent* scale-recursive definition that is vastly simpler to work with and leads to efficient model realization. Because of the structure it reveals and the efficiency to which it leads, the development of this scale-recursive Markov property is one of the important contributions of this thesis.

The efficiency of the realization approach developed in this thesis stems also from the fact that it is based on the estimation-theoretic concept of *predictive efficiency* [8, 154] rather than on canonical correlations. In brief, predictive efficiency is the idea of finding and prioritizing the best (in a minimum mean-square error sense) linear functionals of one random vector for the purpose of linearly estimating another. An important feature of predictive efficiency is the asymmetric way in which it treats data and variables to be estimated. A consequence of this for our approach to the stochastic realization problem is that state variables are chosen to provide maximal *total* reduction in estimation error variance. This is in contrast to the canonical correlations approach which provides maximal *fractional* error variance reduction and is therefore equally concerned with low- and high-variance features. Another important advantage of predictive efficiency's asymmetry is that it avoids the costly inversion and singular value decomposition of large matrices, steps which cannot be avoided in the canonical correlations approach.

■ 1.2.2 Unification of the Wavelet and MAR Frameworks

Like other methods [96,98,100], the approach to the MAR stochastic realization problem summarized in the previous section focuses on designing MAR states to, in some sense, optimally match the statistics of the fine-scale process being modeled. As a consequence, the resulting states typically have no discernible structure beyond the fact that they represent solutions to specific optimization problems. Another approach, which does not suffer from this limitation, is to choose the linear functionals that define the elements of internal MAR states from a library of linear functionals that have some convenient structure. This idea is one motivation for one of the contributions of this thesis—the unification of the wavelet and MAR frameworks. This union proves to be a powerful one because it combines the modeling efficacy of wavelets with the processing efficiency and flexibility of the MAR framework.

MAR processes were motivated by and have much in common with wavelets. In particular, MAR processes and wavelet synthesis both construct signals by adding detail at each successive finer scale. Despite the apparent similarities, however, it seemed, until recently, that the two frameworks could not be easily reconciled except in the simplest case of the Haar wavelet [38, 68]. In this thesis, we show that through a particular definition of MAR state vectors, MAR dynamics can be chosen to match the reconstruction algorithm associated with *any* compactly supported orthogonal or biorthogonal wavelet. Our ultimate objective, however, is not signal synthesis but rather stochastic *modeling*. As mentioned previously, internality is a desirable property of a MAR model because it simplifies MAR model identification.

In the early attempts to marry MAR processes and wavelets, it was incorrectly thought that the internal property doomed the union in all but the Haar case. This is because, for all but the Haar wavelet, the supports of the wavelet functions overlap. We will show the connection between the overlapping of the wavelet functions and the internal property and illustrate how the non-overlapping property of the Haar wavelet permits its simple union with the MAR framework. After proving some particular relationships between wavelet coefficients and through appropriate state augmentation, we show how to build internal MAR processes based on any compactly supported wavelet.

Our main objective, after showing how to unify the MAR and wavelet frameworks, is to build approximate internal MAR models for stochastic processes. To do so, we use the statistics of the process to be modeled to derive the dynamics of our internal MAR-wavelet models. While wavelets have nice decorrelation properties, the decorrelation they provide is not exact in general. Therefore, our MAR models based on wavelets are approximate. This does *not* mean that we assume in our internal models that the detail coefficients are white. In fact, while (for comparison purposes) we do make this assumption for what we shall call the standard MAR-wavelet model, our internal MAR-wavelet models are more sophisticated. In particular, they incorporate the powerful property of optimal stochastic prediction for the detail coefficients at a given scale from both detail *and* scaling coefficients at coarser scales. This is different from the common wavelet-based modeling in which the detail coefficients are assumed to be white. In our internal models, we make the weaker assumption that the *errors* in *predicting* the detail coefficients from coarser-scale coefficients are white.

MAR-wavelet states of low dimension can lead to surprisingly good models. We will see that the state dimension of our MAR-wavelet models grows only linearly with the lengths of the support of the scaling functions, which are related, in some cases, such as orthogonal wavelets, to the number of vanishing moments of the analyzing wavelet. However, the fact that wavelets with a large number of vanishing moments do a good job of whitening a large class of processes [4,72,166,179,199] does *not* imply that the degree of statistical fidelity of our internal models necessarily increases with the number of vanishing moments. This is because we are not exclusively concerned with the correlations between wavelet coefficients, but rather with the *conditional* correlation between them (i.e., the correlation remaining after conditioning on coarser-scale coefficients). Therefore, as will be illustrated, with internal MAR-wavelet models, it is possible to build accurate models using wavelets with fairly short supports and, thus, without dramatically increasing the state dimension.

■ 1.2.3 Covariance Extension

A limitation of all previous approaches to the MAR stochastic realization problem (including those discussed in the preceding two sections) is that they rely on the complete knowledge of the second-order statistics of the signal to be modeled. In many real-world problems, it is unlikely that one will have this complete knowledge. Moreover, in large image (and higher-dimensional) processing problems, such knowledge is impractical due to the amount of memory it would require. Therefore, if the MAR framework is ultimately to be applied to real-world problems of the type just described, it is essential that model identification techniques be developed that do not rely on a complete second-order characterization. A substantial contribution of this thesis is the development of techniques for MAR model realization for circumstances in which the covariance matrix of the signal to be modeled is only partially known.

The problem of inferring the unknown elements of a covariance matrix from known ones is the covariance extension problem which has been extensively studied [2,10,11, 33,51,58,88,102,120,132,159]. One well-known result is that, for certain problems, an efficient recursive algorithm—Levinson's algorithm [21,89,118,119,152]—may be used to compute extensions. One of the contributions of this thesis is to provide a generalization of the classical Levinson algorithm. This generalized-Levinson algorithm, like its classical counterpart, computes a covariance extension one element at a time. Each new element is parameterized by a so-called *reflection coefficient* and, collectively, the reflection coefficients parameterize all valid extensions (i.e., those that are positive semi-definite). The definition of generalized-reflection coefficients, the development of the generalized-Levinson algorithm, and the precise characterization of the problems to which it can be applied are based on the graph-theoretic concept of chordality, which will be explained in detail.

While we will rely and build on the theory associated with covariance extension, we are, ultimately, not interested in finding a full extension of the given, partially known covariance matrix. Rather, our interest is in a MAR *model* for an extension. In particular, for one-dimensional signals, we show how to obtain an exact MAR model for the maximum-entropy extension of a banded, partially known covariance matrix (i.e., correlations among the k nearest neighbors of each point are known). To obtain this model, only a small subset of the unknown covariance elements need to be computed, and they may be computed using the generalized-Levinson algorithm we develop. This results in an order of magnitude of computational savings relative to explicitly computing the full extension. In fact, the complexity of computing the parameters of a MAR model for the maximum-entropy extension of a banded, partially known covariance matrix is equivalent (asymptotically) to computing the parameters of a standard autoregressive model using the classical Levinson algorithm.

■ 1.3 Thesis Organization and Main Contributions

The remainder of this thesis is organized as follows.

Chapter 2, Preliminaries

This chapter begins with a review of linear least-squares estimation with an emphasis on structured models that lead to efficient algorithms. This discussion leads naturally to a consideration of state-space models which are a special case of MAR models. After a formal introduction to MAR processes and their associated signal processing algorithms, the chapter turns to a detailed discussion of previously-developed realization theory. This discussion includes a review of MAR models based on the Haar wavelet which have been used for modeling fractional Brownian motion [37,38,63,68], MAR models for Markov processes [127,129], stochastic realization based on canonical correlations [96,98,100], the overlapping tree approach to achieve smoothness [63,96,97], and a state augmentation technique for incorporating nonlocal variables [37,39]. The chapter concludes with a review of predictive efficiency [8,154], a technique which we will use in Chapter 4 for MAR model realization.

Chapter 3, Internality and Markovianity

This chapter resolves the issue of MAR model inconsistency which is intimately related to the notion of internality. While this issue has been recognized by previous authors, it has, until now, not been addressed in a theoretically complete and computationally tractable way. The approach developed in this chapter requires the consideration of general tree-indexed processes, not necessarily MAR ones. The theory of internal tree-indexed processes is formally developed and results in a new parameterization for internal processes which leads to MAR models that are consistent. After the development of internality, we consider its consequences for Markovianity and show that it substantially simplifies this important property. The chapter closes with a discussion of the implications of internality and the scale-recursive notion of Markovianity to which it leads for stochastic realization algorithms and their computational complexity.

Chapter 4, Scale-Recursive Stochastic Realization

This chapter develops a MAR stochastic realization algorithm. This algorithm makes use of the theory of internality and scale-recursive Markovianity as developed in Chapter 3. Additionally, it is based on predictive efficiency which is introduced in Chapter 2. These three ideas (internality, scale-recursive Markovianity, and predictive efficiency) combine to result in a realization algorithm that has complexity quadratic in problem size. An approximation is then introduced that leads to a complexity linear in problem size. An analysis of this approximation is provided and the chapter concludes with examples that illustrate the degree of approximation error relative to the exact algorithm.

Chapter 5, MAR-Wavelet Processes

This chapter unifies the wavelet framework and the MAR framework. While MAR processes based on the Haar wavelet have been developed and applied by previous authors (see Chapter 2), generalization to other wavelets has been difficult. This difficulty stems from the constraints imposed by internality, constraints that were not well understood until this thesis (see Chapter 3 and Chapter 4). After proving some particular properties of wavelets, internal MAR processes based on *any* compactly supported orthogonal or biorthogonal wavelet are developed. The chapter concludes with an application of these MAR-wavelet processes to stochastic realization.

Chapter 6, Covariance Extension

All previous systematic approaches to MAR stochastic realization have required complete knowledge of the second-order statistics of the process being modeled. The work presented in this chapter takes a first step toward relaxing this assumption. The chapter includes a detailed review of covariance extension and its relation to graph theoretic concepts. A generalized-Levinson algorithm is developed that permits computation of extensions under quite general conditions. This algorithm is then applied to the problem of designing a MAR model for the maximum-entropy extension of a banded, partially known covariance matrix. A significant feature of maximum-entropy covariance extension using MAR models is that the complexity of the approach is an order of magnitude below that of explicitly computing a full covariance extension. Moreover, the complexity is comparable to that required to build a standard autoregressive model using the classical Levinson algorithm.

Chapter 7, Incorporation of Nonlocal Variables

This chapter presents several methods for incorporating nonlocal linear functionals into MAR models. Such linear functionals may represent coarse-scale measurements or variables to estimate and including them requires care because of the constraints imposed by internality and Markovianity. Three approaches to this problem are developed and they represent powerful alternatives to the approach of [37, 39] which is reviewed in Chapter 2. The first approach incorporates nonlocal linear functionals approximately. The second does so exactly but, in contrast to the method of [37, 39], they are incorporated prior to model realization so that the information they carry may be exploited in modeling signal statistics. Finally, an intellectual successor to the Markov property is developed that focuses only on MAR states that carry information relevant for the estimation problem at hand. This is in contrast to the usual Markov property that focuses on all MAR states regardless of their importance for estimation.

Chapter 8, Contributions and Suggestions

This chapter summarizes the contributions of this thesis and provides suggestions for extending the theoretical foundation developed in the preceding chapters. Other open problems associated with the MAR framework are also discussed.

Chapter 2

Preliminaries

THIS chapter primarily focuses on MAR processes, associated signal processing algorithms, and previously-developed realization theory. The MAR framework, which we review in Section 2.2, was principally developed to support efficient linear least-squares estimation. To motivate the MAR framework and to provide additional insight into its structure, we first provide, in Section 2.1, a review of linear least-squares estimation with an emphasis on structured models that lead to computational efficiencies. Much of the chapter is devoted to MAR realization theory, elements of which are reviewed in Section 2.3. The chapter concludes with Section 2.4 in which we review predictive efficiency, a technique which we will use in Chapter 4 for MAR model realization.

■ 2.1 Linear Estimation and Structured Models

In this section we first consider a generic, finite-dimensional, linear least-squares estimation problem and then consider several classes of such problems which have special structure. Many of the topics covered here are discussed in greater detail in [37, 63, 96, 126, 148, 168, 184, 188]. A finite-dimensional, linear estimation problem is one of estimating a zero-mean[1] vector of unknowns f with a linear function of a zero-mean vector of observations g. That is,

$$\widehat{f}_L = Lg \tag{2.1}$$

is a linear estimate of f. The estimation error is $e_L \triangleq f - \widehat{f}_L = f - Lg$ and the estimation error covariance matrix is

$$P_{e_L} = P_f + LP_gL^T - LP_{gf} - P_{fg}L^T \tag{2.2}$$

where $P_x \triangleq \mathrm{E}[xx^T] - \mathrm{E}[x]\,\mathrm{E}[x]^T$ is our notation[2] for the covariance matrix of random vector x and $P_{xy} \triangleq \mathrm{E}[xy^T] - \mathrm{E}[x]\,\mathrm{E}[y]^T$ is our notation for the cross-covariance matrix for random vectors x and y.

[1] The assumption of zero-mean results in no loss of generality because, for the case of non-zero mean, we may consider the deviation from the mean. This results in an affine estimator rather than a linear one.

[2] $\mathrm{E}[\cdot]$ is the expectation operator. All notational conventions are summarized on pages 17–19.

■ 2.1.1 Linear Least-Squares Estimation

The linear estimator that minimizes the mean-square estimation error is called the linear least-squares (LLS) estimator. The LLS estimate of f based on g will be denoted by $\widehat{\mathrm{E}}[f|g]$ or, when the dependence on g is clear, simply $\widehat{f}$. The form of the LLS estimator and estimation error covariance is

$$\widehat{f} = \widehat{\mathrm{E}}[f|g] = P_{fg}P_g^{-1}g\,, \tag{2.3a}$$

$$P_e = P_f - P_{fg}P_g^{-1}P_{fg}^T \tag{2.3b}$$

where $e = f - \widehat{f}$. The LLS estimator possesses a number of theoretically and practically useful properties including the following.

- The LLS estimator and error covariance depends only on the joint second-order statistics of f and g.
- The LLS estimate is unbiased.
- The LLS estimation error is orthogonal to all affine functions of the data upon which it is based.
- An estimator that is orthogonal to all affine functions of the data upon which it is based is the LLS estimator.
- $P_e - P_{e_L} \leq 0$, where P_e is the LLS estimation error covariance and P_{e_L} is the error covariance of any linear estimator Lg. Equality obtains if and only if Lg is the LLS estimator and inequality is in the sense of negative-definiteness.
- $P_{e|g} = P_e$ where e is the LLS estimation error. That is, the conditional estimation error covariance does not depend on the data g.

A special and important class of LLS estimation problems are those based on an affine measurement model,

$$g = Hf + \nu \tag{2.4}$$

where H is an $N_g \times N_f$ dimensional matrix. When ν is zero-mean, uncorrelated with f, and has a covariance P_ν, the LLS estimator and error covariance have the form

$$\widehat{f} = P_f H^T(HP_fH^T + P_\nu)^{-1}g\,, \tag{2.5a}$$

$$P_e = P_f - P_fH^T(HP_fH^T + P_\nu)^{-1}HP_f\,. \tag{2.5b}$$

Another way of expressing (2.5) is

$$\widehat{f} = (P_f^{-1} + H^TP_\nu^{-1}H)^{-1}H^TP_\nu^{-1}g\,, \tag{2.6a}$$

$$P_e^{-1} = P_f^{-1} + H^TP_\nu^{-1}H\,. \tag{2.6b}$$

We point out for future reference that (2.6a) may be written as

$$P_e^{-1}\widehat{f} = H^T P_\nu^{-1} g\,. \tag{2.7}$$

The computational complexity of implementing (2.5) or (2.6) depends on N_f and N_g, the dimensions of the unknown vector and the data vector, respectively. For the case in which P_ν is diagonal but all other matrices in (2.5) and (2.6) are full, direct solution (i.e., with explicit matrix inversions) of (2.5) requires $O(N_g^3 + N_f^2 N_g)$ computations and (2.6) requires $O(N_f^3)$ computations. Thus, when $N_g \ll N_f$, (2.5) will be more efficient to implement, and when $N_f \ll N_g$, (2.6) will be more efficient to implement. In either case, the computational complexity of implementing (2.5) or (2.6) becomes prohibitively burdensome as N_g and N_f become large. Additionally, explicit storage of P_e is prohibitive for large N_f.

If one requires only the estimates and not the estimation error covariance then standard conjugate gradient-based methods [52,84] can be used to solve (2.7) iteratively and, in many cases, with $O(N_f)$ complexity. Recently, a new Krylov subspace method has been developed [161,163,164] that extends the standard conjugate gradient methods and iteratively computes both the estimates and the diagonal elements of the estimation error covariance (i.e., the estimation error variances) with, asymptotically, no more work than standard conjugate gradient. Moreover, this technique yields an implicit and approximate model for the entire error covariance matrix. This approach is applicable in cases where the matrix-vector product $P_g z$ can be computed quickly for any length-N_g vector z and where the estimator can be interpreted as acting as a low-pass filter.

■ 2.1.2 Structured Models

Computational savings can be achieved in some cases for which the matrices in (2.5) and (2.6) have additional structure. One such case is when H is the identity matrix and f is stationary with circular boundary conditions (i.e., in one dimension, f is indexed by a circular lattice and in two dimensions, a toroidal one). In this case, all of the matrices that appear in (2.5) can be diagonalized by the discrete Fourier transform basis vectors and the diagonalization may be implemented efficiently with the fast Fourier transform (FFT) [96,182]. Hence, estimates and error statistics may be computed with complexity $O(N_f \log(N_f))$. Other fast transform techniques, like the wavelet transform, lead to efficient estimation and the computation of error statistics in some cases in which f is nonstationary [191]. However, one drawback to fast transform techniques (FFT or otherwise) is that they require that $H = I$, corresponding to a dense, regularly-spaced set of measurements. Another is that they require that the statistics possess special structure (e.g., stationarity in the case of the Fourier transform).

Another type of structured LLS problem that leads to some level of computational savings arises when f is a wide-sense Markov random field (MRF). In this case, f can be specified implicitly as [53,121,122,189]

$$Mf = \eta \tag{2.8}$$

where M is a symmetric matrix representing a discretization of an elliptic partial differential operator and, so, is sparse. Moreover, M is the covariance matrix for η and, hence, the inverse of the covariance matrix for f (i.e., $P_f^{-1} = M$). If the measurements are point observations (i.e., the rows of H are a subset of the rows of the identity matrix) then $H^T P_\nu^{-1} H$ is diagonal, assuming that P_ν is diagonal. Therefore, (2.7) is a sparse system of equations where

$$P_e^{-1} = P_f^{-1} + H^T P_\nu^{-1} H = M + H^T P_\nu^{-1} H \tag{2.9}$$

has the same sparsity structure as M.

While there are a variety of iterative procedures one can apply to MRF problems corresponding to sparse systems of equations of the type just described (e.g., relaxation and multigrid methods [20,178]), these do not produce estimation error statistics. Nested dissection [57,61,82,187], on the other hand, *can* be used to compute directly *both* the estimates with $O(N_f^{3/2})$ complexity and, with only a little extra work, the elements of P_e that lie in the non-zero locations of a Cholesky factor of M. A significant limitation of nested dissection, however, is that it cannot efficiently handle a substantial number of nonlocal measurements as they destroy the sparsity of P_e^{-1}.

Having just discussed special classes of two-dimensional processes, we now turn to one-dimensional ones. One-dimensional, wide-sense Markov processes can be written as state-space processes, which are a sub-class of MAR processes. A state-space process and corresponding observation equation have the form

$$x(n) = A(n)x(n-1) + w(n)\,, \tag{2.10a}$$
$$y(n) = C(n)x(n) + v(n) \tag{2.10b}$$

where $x(\cdot)$, $w(\cdot)$, and $v(\cdot)$ are all zero-mean, $w(\cdot)$ is white, uncorrelated with $x(0)$ (the assumed initial value of the Markov process), $v(\cdot)$ is white, uncorrelated with $x(\cdot)$, $w(\cdot)$. If we define

$$f \triangleq \begin{bmatrix} x(0)^T & x(1)^T & \cdots & x(N-1)^T \end{bmatrix}^T , \tag{2.11a}$$
$$\eta \triangleq \begin{bmatrix} x(0)^T & w(1)^T & \cdots & w(N-1)^T \end{bmatrix}^T \tag{2.11b}$$

then it is clear that (2.10a) can be written in the form of (2.8) with

$$M = \begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ -A(1) & I & 0 & \cdots & 0 & 0 \\ 0 & -A(2) & I & \cdots & 0 & 0 \\ 0 & 0 & -A(3) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -A(N-1) & I \end{pmatrix} . \tag{2.12}$$

However, in contrast to the MRF case, for a state-space process η is white and so $P_f^{-1} = M^T M$ and is block tri-diagonal. A similar correspondence can be made between

(2.10b) and (2.4) in which H is block diagonal. Hence, (2.7) is a sparse system where

$$P_e^{-1} = P_f^{-1} + H^T P_\nu^{-1} H = M^T M + H^T P_\nu^{-1} H \tag{2.13}$$

is block tri-diagonal.

State-space processes are statistically rich and admit efficient inference algorithms. Indeed, any wide-sense stationary process with a rational spectrum and many nonstationary processes can be written in the form of a state-space process. The Kalman filter [104, 105] and Rauch-Tung-Striebel smoother [155] may be used to solve for the LLS estimates and estimation error variances. The former corresponds to Gaussian elimination on the system (2.7), and the latter corresponds to back substitution. Due to the tri-diagonal structure of P_e^{-1}, these algorithms are extremely efficient, with complexity linear in N, the temporal length of the state-space process, and cubic in the dimension of the state vectors $x(\cdot)$. Moreover, in solving for the LLS estimates, these algorithms produce, as a by-product, estimation error variances as well as a model for the estimation error which is, itself, a state-space process. These algorithms may also be used to whiten the data $y(\cdot)$ and, thus, admit efficient computation of likelihood functions.

State-space processes are indexed by the integers. Therefore, they constitute a natural modeling framework for one-dimensional signals. While they have been applied to two-dimensional problems, doing so efficiently requires the imposition of a somewhat unnatural and ad hoc ordering of image pixels (e.g., raster-scan ordering). A natural question to pose is: how might one retain the attractive features of state-space processes while obviating the inflexibility of integer indexing?

The answer is to generalize the index set from integers (which form a monadic tree) to trees. This generalization gives rise to MAR processes which evolve, not in time, but in *scale*. As we review in the following sections, there is no loss in statistical richness or algorithmic efficiency associated with the generalization of state-space processes to MAR processes. Indeed, the class of processes that can be modeled with MAR models *includes* state-space processes and the MAR inference algorithms are generalizations of state-space ones.

■ 2.2 MAR Framework

In this section we provide our notational conventions for trees and tree-indexed processes as well as a review of MAR processes and associated statistical signal processing algorithms. In this thesis, we will consider processes indexed by q-adic trees, although all the techniques developed are applicable to processes indexed by any tree, however irregular. Our notation for referring to nodes of a q-adic tree is indicated in Figure 2.1. The root node is labeled 0, the parent of node s is denoted by $s\bar{\gamma}$, and the children of node s are, from left to right, $s\alpha_1, s\alpha_2, \ldots, s\alpha_q$.

There is a natural notion of scale associated with q-adic trees. The root node represents the coarsest scale (scale zero), while the leaf nodes constitute the finest scale

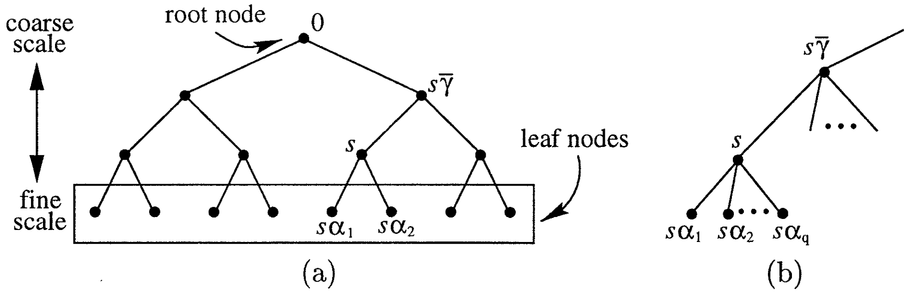


Figure 2.1. (a) A dyadic tree. The root node is indexed by $s = 0$. The parent of node s is denoted $s\bar{\gamma}$. The children of node s are labeled from left to right by $s\alpha_1$, $s\alpha_2$. (b) For a q-adic tree the children of node s are labeled from left to right by $s\alpha_1, s\alpha_2, \ldots, s\alpha_q$.

(scale M). More generally, the nodes $\{s \mid s\bar{\gamma}^n = 0\}$ reside at scale n. We denote the scale of node s by $m(s)$. The shift of node s is denoted by $\imath(s)$ where the left-most node at a given scale has shift $\imath(s) = 0$, and the right-most node has shift $\imath(s) = q^{m(s)} - 1$. For dyadic trees, $q = 2$ and scale n indexes a one-dimensional vector-valued signal of length 2^n. For quad-trees, $q = 4$ and scale n indexes a two-dimensional vector-valued field of size $2^n \times 2^n$. Extensions to other values of $q > 4$ are straightforward.

A MAR process is a generalization of a discrete-time state-space process. Both are graphical models with affine dynamics. However, a MAR process may be indexed by the nodes of *any* tree[3] and it reduces to a state-space process in time when the tree is monadic. Precisely, a zero-mean MAR process $x(\cdot)$ has dynamics

$$x(s) = A(s)x(s\bar{\gamma}) + w(s) \tag{2.14}$$

where $w(s)$ is white with auto-covariance $Q(s)$ and is uncorrelated with $x(0)$, the value of the MAR process at the root node. With these definitions, it is clear that (2.14) is a generalization of the state-space dynamics of (2.10a). In analogy with state-space processes and for other reasons that will be made clear in the sequel, $x(s)$ is called the *state* of $x(\cdot)$ at node s.

In this thesis, we will provide techniques for building MAR models for fine-scale random signals which we view as indexed by the leaf nodes of q-adic trees. In our development of theory and algorithms, we frequently refer to other scales and other subsets of nodes. So, for simplicity of our subsequent presentation, we make the following definitions for subsets of the set of nodes of a q-adic tree to which we refer

[3] Strictly speaking, a MAR process may be indexed by the nodes of any *directed* tree where the edges between nodes are directed from parent to child (coarse-scale to fine-scale). Our notation for nodes (s, $s\bar{\gamma}$, $s\alpha_i$) and the orientation of our figures (coarser scales are above finer ones) make this sense of direction clear so our figures are drawn with undirected (i.e., arrow-less) edges.

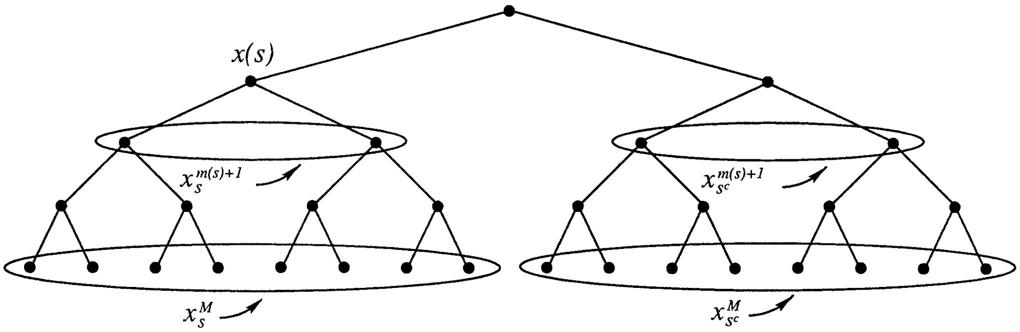


Figure 2.2. Notation for sub-processes of a tree-indexed process $x(\cdot)$.

frequently:

$$\begin{aligned}
\mathcal{S}_s &\triangleq \{t \mid t = s \text{ or } t \text{ is a descendent of } s\} = \text{nodes in subtree rooted at } s\,,\\
\mathcal{S}_s^c &\triangleq \mathcal{S}_0 - \mathcal{S}_s = \text{nodes other than those in subtree rooted at } s\,,\\
\mathcal{T}_s(n) &\triangleq \{t \in \mathcal{S}_s \mid m(t) = n\} = \text{nodes at scale } n \text{ descending from } s\,,\\
\mathcal{T}_s^c(n) &\triangleq \mathcal{T}_0(n) - \mathcal{T}_s(n) = \text{nodes at scale } n \text{ not descending from } s\,.
\end{aligned}$$

Again, to simplify our development, we make the following definitions for sub-processes of a tree-indexed process $\{x(s)\}_{s\in\mathcal{S}_0}$ to which we refer frequently:

$$\begin{aligned}
x_s^n &\triangleq \{x(t)\}_{t\in\mathcal{T}_s(n)} = \text{process at scale } n \text{ that descends from node } s\,,\\
x_{s^c}^n &\triangleq \{x(t)\}_{t\in\mathcal{T}_s^c(n)} = \text{process at scale } n \text{ that does not descend from node } s\,.
\end{aligned}$$

We often interpret these sub-processes as vectors.[4] Also, when referring to the entire sub-process at a particular scale we often drop the 0 subscript. For instance $x^M \equiv x_0^M$ is the finest-scale sub-process. Some of this notation is summarized in Figure 2.2.

MAR dynamics provide a complete and implicit second-order characterization for the collection of states $\{x(s)\}_{s\in\mathcal{S}_0}$. The state covariances satisfy the Lyapunov equation

$$P_{x(s)} = A(s)P_{x(s\bar{\gamma})}A(s)^T + Q(s) \tag{2.15}$$

which is a generalization of the more familiar state-space Lyapunov equation. The cross-covariance between any two states is most easily written in terms of the state transition matrix[5]

$$\Phi(s,t) = \begin{cases} I & \text{if } s = t\,,\\ A(s)\Phi(s\bar{\gamma},t) & \text{if } m(s) > m(t)\,,\\ \Phi(s,t\bar{\gamma})A(t)^T & \text{if } m(t) > m(s)\,. \end{cases} \tag{2.16}$$

[4] Using, for example, lexicographic ordering of the nodes comprising $\mathcal{T}_s(n)$ or $\mathcal{T}_s^c(n)$ in order to construct a large vector from the component vectors $x(t)$.

[5] Note that $\Phi(s,t)$ is only defined for nodes s,t such that s is an ancestor of t or vice versa. That is, $s = t\bar{\gamma}^\ell$ or $t = s\bar{\gamma}^\ell$ for some non-negative integer ℓ.

which is a generalization of the more familiar state transition matrix for state-space processes. The cross-covariance between any two states $x(t)$ and $x(s)$ is

$$P_{x(s)x(t)} = \Phi(s, s \wedge t) P_{x(s \wedge t)} \Phi(s \wedge t, t) \tag{2.17}$$

where $s \wedge t$ is defined as the common ancestor of s and t with maximal scale. A consequence of the foregoing discussion is that the MAR dynamics can be viewed as an implicit representation of P_{x^M}, the covariance matrix for the leaf-node states of a MAR process.

The most important property that MAR processes possess is a wide-sense Markovianity.

Definition 2.2.1 (Markov Property). *A tree-indexed process $x(\cdot)$ has the* Markov property *if for all $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$, conditioned on $x(s)$, the sub-processes indexed by the sets of nodes in the sub-trees separated by s, namely, $\{x(t)\}_{t \in \mathcal{S}_{s\alpha_1}}$, $\{x(t)\}_{t \in \mathcal{S}_{s\alpha_2}}, \ldots,$ $\{x(t)\}_{t \in \mathcal{S}_{s\alpha_q}}$ and $\{x(t)\}_{t \in \mathcal{S}_s^c}$, are conditionally uncorrelated.*

That MAR processes have the Markov property is easily shown [37, Appendix A]. If a MAR process is Gaussian then it has the (equivalent) properties of "pairwise Markovianity," "local Markovianity," and "global Markovianity" from the graphical modeling literature [115]. The Markov property of Definition 2.2.1 is a wide-sense equivalent to these notions of Markovianity.

The Markov property leads to fast statistical signal processing algorithms. Sample-path generation is accomplished using (2.14) and has complexity $O(\sum_{s \in \mathcal{S}_0} d_s^2)$ where d_s is the dimension of $x(s)$. Also, a linear least-squares estimator [28–30] (which generalizes the Kalman filter [104, 105] and Rauch-Tung-Striebel smoother [155]) and likelihood calculator [127, 130] have been developed based on a measurement model analogous to the classical state-space one:

$$y(s) = C(s)x(s) + v(s) \tag{2.18}$$

where $v(s)$ is white and uncorrelated with $x(\cdot)$ and $w(\cdot)$. The estimator and likelihood calculator have computational complexity $O(\sum_{s \in \mathcal{S}_0} d_s^3)$. When the state dimension is a constant d which is independent of $|\mathcal{S}_0|$, the complexity of the sample-path generation and estimation/likelihood calculation is $O(d^2|\mathcal{S}_0|)$ and $O(d^3|\mathcal{S}_0|)$, respectively. Since $|\mathcal{S}_0|$ is an affine function of the number of fine-scale variables, $N = dq^M$, this corresponds to a constant per pixel, or $O(N)$, complexity.[6]

We emphasize that this efficiency is only achieved if the state dimension, d, is independent of $|\mathcal{S}_0|$ or, equivalently, N. If d grows slowly with N, say, logarithmically, then reasonably low complexity can also be achieved. However, if d is proportional to N or even $N^{1/2}$, the complexity of these signal processing algorithms may be prohibitive for large problems. While there are many examples of processes for which the state dimension is independent of, or grows slowly with, N (e.g., one-dimensional Markov processes,

[6] For a q-adic tree, $|\mathcal{S}_0| = q^{M+1} - 1$. Since $N = dq^M$, we have that $|\mathcal{S}_0| = q(\frac{N}{d}) - 1$ which is affine in N.

$1/f$-like processes), there is no theory that completely answers the question: what class of processes can be well-modeled in the MAR framework with state dimensions that grow sufficiently slowly with problem size?

In addition to computational efficiency, there are a number of other advantages to estimation within the MAR framework. One is statistically optimal fusion of multiresolution data. Indeed, since the MAR estimator computes the LLS estimate of $x(\cdot)$ given $y(\cdot)$ *for all* $s \in \mathcal{S}_0$, it automatically fuses data at coarse nodes (which may represent nonlocal measurements) with those at leaf nodes. This feature distinguishes the MAR framework from nested dissection which cannot efficiently handle nonlocal measurements.

A second advantage is that the MAR estimator can handle irregularly spaced data and nonstationary statistics *with no algorithmic changes.* That is, the algorithm, much like the Kalman filter and Rauch-Tung-Striebel smoother on which it is based, does not require any additional structure such as regularly spaced data or stationarity to achieve its efficiency. This is in contrast to fast-transform techniques such as those based on the FFT or wavelet transform which require either regularly spaced data or stationarity, or both.

A third important point is that the MAR estimator produces estimation error variances *with no additional computations beyond those that are needed to compute the estimates themselves.* This fact distinguishes the MAR framework from other approaches (like multigrid) that lead to fast algorithms for estimates but which cannot efficiently produce error statistics.

Lastly, the MAR estimator produces the parameters of a MAR model for the estimation errors with only a small amount of additional work beyond what is needed to compute estimates [127, 131]. The MAR error model can be used compute individual off-diagonal elements of the estimation error covariance. Moreover, the error model is useful for generating conditional sample-paths [37, 41], for performing sequential data fusion, or for adapting the MAR estimator to time-dynamic problems [90–92].

■ 2.3 MAR Stochastic Realization Theory

The MAR stochastic realization theory developed in this thesis is motivated by and, to some extent, directly builds upon the prior work of others [37–39, 63, 68, 96–98, 100, 127, 129]. In this section we review this previously-developed realization theory which primarily addresses the following problem. Suppose we are given the covariance matrix P_{f^M} for the length N random vector f^M which may represent a one-dimensional signal or a multi-dimensional field, lexicographically ordered. The stochastic realization problem is to specify the parameters of a MAR process $x(\cdot)$ so that the finest-scale subprocess x^M is an exact or approximate model for the random signal f^M, i.e., so that $P_{x^M} \approx P_{f^M}$.

The parameters, $A(\cdot)$, $Q(\cdot)$ and $P_{x(0)}$, that govern MAR dynamics are a function of state covariances $P_{x(s)}$ and child-parent cross-covariances, $P_{x(s)x(s\bar{\gamma})}$. To see this, notice that the MAR dynamics of (2.14) represent the linear least squares estimate of $x(s)$ from $x(s\bar{\gamma})$ plus the estimation error $w(s)$. Therefore,

$$A(s) = P_{x(s)x(s\bar{\gamma})} P^{-1}_{x(s\bar{\gamma})} , \tag{2.19a}$$

$$Q(s) = P_{x(s)} - P_{x(s)x(s\bar{\gamma})} P^{-1}_{x(s\bar{\gamma})} P^{T}_{x(s)x(s\bar{\gamma})} . \tag{2.19b}$$

Thus, to specify a MAR process, we need only determine the joint second-order statistics of all child-parent pairs. As developed in [96, 98, 100], this task is vastly simplified by considering a so-called *internal* MAR process[7] which is one with the property that each state $x(s)$ is linearly related to x^M, the vector of states indexed by fine-scale nodes. That is, for all $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$,

$$x(s) = L_s x^M \tag{2.20}$$

for some set of matrices $\{L_s\}$. A point recognized, but not resolved, in [96, 100] is that (2.20) is *not* an appropriate parameterization for internal MAR states as it can lead to inconsistent models. One of the main contributions of this thesis is to resolve this issue of model consistency.

If our MAR model is exact, then $P_{x^M} \equiv P_{f^M}$ and the state covariances and child-parent cross-covariances are given by

$$P_{x(s)} = L_s P_{x^M} L_s^T = L_s P_{f^M} L_s^T , \tag{2.21a}$$

$$P_{x(s)x(s\bar{\gamma})} = L_s P_{x^M} L_{s\bar{\gamma}}^T = L_s P_{f^M} L_{s\bar{\gamma}}^T . \tag{2.21b}$$

As discussed in [96, 98, 100], (2.21) may be used to define $P_{x(s)}$ and $P_{x(s)x(s\bar{\gamma})}$ even when $P_{x^M} \neq P_{f^M}$. However, a clear interpretation of this step is not provided in [96, 98, 100] and another contribution of this thesis is to make this step precise. Despite this imprecision and the consistency issues mentioned previously, (2.19)–(2.21) correctly imply that the stochastic realization problem can be posed as a problem of finding a set of linear functionals $\{L_s\}$ that define MAR states. In the remainder of this section we discuss several ways of specifying these linear functionals and the consequences for stochastic realization.

Each of the following five sections provides the background necessary for subsequent developments. However, a reader interested in only certain specific topics need not read all five sections. Table 2.1 serves as a guide and indicates to which contribution of this thesis each of the following sections pertains and in what section or chapter the development of that contribution may be found.

■ 2.3.1 MAR-Haar Models for Fractional Brownian Motion

Perhaps the quickest route toward understanding MAR stochastic realization begins with examining a conceptually simple example. In this section we will do just that

[7] The property of internality will be developed in *much* greater depth in Chapter 3.

Section	*Background for*	*Found in*
2.3.1	MAR-wavelet unification	Chapter 5
2.3.2	covariance extension	Chapter 6
2.3.3	predictive efficiency	Section 2.4
2.3.4	examples	Section 4.3
2.3.5	incorporation of nonlocal variables	Chapter 7

Table 2.1. Guide to Section 2.3.

by considering MAR processes based on the Haar wavelet. Our discussion is based on ideas in [37, 38, 44, 63, 68] and we refer the reader to these papers for more detail.

The Haar *scaling function*[8] is

$$\phi(t) = \begin{cases} 1 & \text{for } 0 \le t < 1\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.22}$$

For a fixed scale j, the family of functions given by $\phi_{j,n} \triangleq \sqrt{2^j}\phi(2^j t - n)$ for $n \in \mathbb{Z}$ constitutes an orthonormal basis for functions that are piecewise constant on intervals of length 2^{-j}. Therefore, with increasing j, $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ is a basis for finer resolution function approximation. Expanding a function in this basis we obtain for each j, $\sum_n a_j(n)\phi_{j,n}$ where the *scaling coefficients* $\{a_j(n)\}_{(j,n)\in\mathbb{Z}^2}$ associated with the scaling functions $\{\phi_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ obey the fine-to-coarse scale-recursive *analysis equation*

$$a_j(n) = \frac{1}{\sqrt{2}}a_{j+1}(2n) + \frac{1}{\sqrt{2}}a_{j+1}(2n+1)\,. \tag{2.23}$$

Note that, while our convention of finer resolution corresponding to increasing j is consistent with the indexing of MAR scales, it is opposite of the customary convention adopted in the wavelet literature.

The detail required to obtain the j-th resolution approximation of a function from the approximation at resolution $j-1$ is given in terms of the Haar *wavelet function*:

$$\psi(t) = \begin{cases} 1 & \text{for } 0 \le t < 1/2\,, \\ -1 & \text{for } 1/2 \le t < 1\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.24}$$

The family of functions indexed by $(j,n) \in \mathbb{Z}^2$ and given by $\psi_{j,n} \triangleq \sqrt{2^j}\psi(2^j t - n)$ constitutes an orthonormal wavelet basis of $L^2(\mathbb{R})$. The *detail coefficients* $\{d_j(n)\}_{(j,n)\in\mathbb{Z}^2}$ associated with the wavelet functions $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ obey the fine-to-coarse scale-recursive analysis equation

$$d_j(n) = \frac{1}{\sqrt{2}}a_{j+1}(2n) - \frac{1}{\sqrt{2}}a_{j+1}(2n+1)\,. \tag{2.25}$$

[8] A more comprehensive review of wavelets is found in Chapter 5.

The Haar *synthesis equation* obeys the coarse-to-fine scale-recursion

$$a_j(n) = \frac{1}{\sqrt{2}} a_{j-1}(\lfloor n/2 \rfloor) + \frac{(-1)^n}{\sqrt{2}} d_{j-1}(\lfloor n/2 \rfloor) \tag{2.26}$$

where $\lfloor \cdot \rfloor$ is the "floor" function and denotes the greatest integer that is no larger than the argument. Thus, for the Haar system, each scaling coefficient $a_j(n)$ for the j-th resolution approximation is derived from *one* scaling coefficient $a_{j-1}(\lfloor n/2 \rfloor)$ for the approximation at resolution $j-1$ and *one* detail coefficient $d_{j-1}(\lfloor n/2 \rfloor)$. Using this fact and assuming that the detail coefficients are white noise, (2.26) can be easily written as a MAR process, as shown in [63, 68], by defining each $x(s)$ as containing a scaling *and* detail coefficient for scale $m(s)$ and shift $\imath(s)$. That is, let $x(s) \triangleq \left[a_{m(s)}(\imath(s)) \quad d_{m(s)}(\imath(s))\right]^T$. The MAR dynamics for such a model are given by

$$x(s) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & (-1)^{\imath(s)} \\ 0 & 0 \end{bmatrix} x(s\bar{\gamma}) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix} d_{m(s)}(\imath(s))}_{w(s)} \tag{2.27a}$$

for scale $m(s) < M$ and

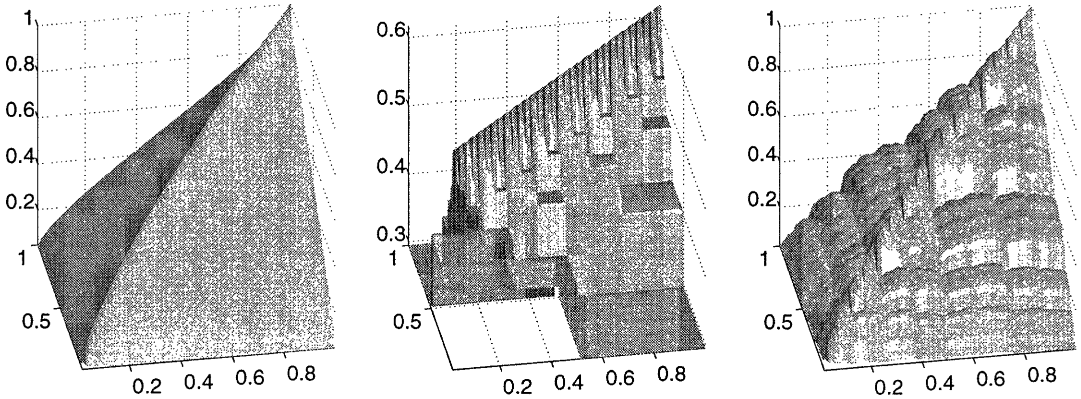
$$x(s) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & (-1)^{\imath(s)} \end{bmatrix} x(s\bar{\gamma}) \tag{2.27b}$$

for scale $m(s) = M$. Notice that (2.27a) is simply a rewriting of (2.26) and (2.27b) terminates the scale-recursive synthesis at scale M.

To use the MAR-Haar process just defined as a model for a random process f^M with covariance P_{f^M} we must specify the covariance matrix for $x(0)$ and the variance of the detail coefficients $d_j(n)$ which act as driving noise in (2.27a). This is easy to do if we view the state elements of the MAR-Haar process as consisting of scaling and detail coefficients of f^M. Indeed, then $x(0) = L_0 f^M$ where the matrix L_0 is implicitly given by recursive application of (2.23) and (2.25). Similarly, the $d_j(n)$ are linearly related to f^M via (2.25). Since P_{f^M} is given, we may immediately derive the required statistics. Notice that we *do not* need to derive the matrices $A(\cdot)$ because they are specified by the Haar synthesis equation as indicated in (2.27a) and (2.27b). The simplicity of this procedure (originally described in [63, 68]) belies a major weakness—it leads to an inconsistent model. This issue will be further developed in the subsequent chapters of this thesis.

As an example, consider using the MAR-Haar process as a model for fractional Brownian motion with Hurst parameter H (fBm(H)). Fractional Brownian motion has received a great deal of attention in recent years [3, 70–72, 113, 179, 190–193] due to the fact that it captures the $1/f$ spectral behavior of many natural and engineered phenomena [17, 109, 124, 125, 185, 195]. The correlation function for fBm(H) is [135]

$$r_H(t_1, t_2) = \frac{1}{2}\left(|t_1|^{2H} + |t_2|^{2H} - |t_1 - t_2|^{2H}\right). \tag{2.28}$$



(a) (b) (c)

Figure 2.3. (a) Exact covariance for 64 samples of fBm(0.3) on $(0, 1]$. (b) Realized covariance using the MAR-Haar model of (2.27a) and (2.27b). (c) Realized covariance using the MAR-Haar model with optimal prediction from parent to child.

Figure 2.3(a) illustrates P_{f^M}, 64 samples of fBm(0.3) on the interval $(0, 1]$. Figure 2.3(b) illustrates the realized fine-scale covariance, P_{x^M}, based on the MAR-Haar model just described. It is immediately clear from Figure 2.3(a) and Figure 2.3(b) that the MAR-Haar model is a crude approximation to the exact fBm(0.3) statistics. The approximation stems from two sources. First, the assumption that the detail coefficients are white is very poor in general. Indeed, since the Haar wavelet has only one vanishing moment, for most processes the resulting detail coefficients are strongly correlated both in space and scale [72, 179]. Second, due to the piecewise constant shape of the Haar wavelet, any realized covariance matrix P_{x^M} with this model will exhibit discontinuities (i.e., will have "blockiness") in general. We point out however, that this model has been successfully used for hypothesis discrimination. In [63, 68] the authors applied the MAR likelihood calculator to estimate accurately the Hurst parameter of fBm.

A simple way to improve upon the MAR-Haar model is to exploit fully the fact that states are linearly related to the fine-scale process. Instead of using the dynamics defined by (2.27a) and (2.27b), one can build a more accurate model, as shown in [37,38], by computing the MAR parameters so that the multiscale autoregression is the optimal prediction of $x(s)$ from $x(s\bar{\gamma})$. That is, using the fact that $x(s) = L_s x^M$, we can directly apply (2.19) and (2.21). Thus, while $w(s)$ in the model defined by (2.27a) and (2.27b) represents the detail coefficient $d_{m(s)}(\imath(s))$, the process noise in the more accurate model we are now considering represents the prediction error in the estimation of $d_{m(s)}(\imath(s))$ *conditioned* on the detail and scaling coefficient represented by $x(s\bar{\gamma})$.

This new and more accurate multiscale model will capture correlations in scale among the coefficients represented by states at neighboring nodes in the tree. Thus, it will do a better job of approximating the statistics of the underlying process than does the model defined by (2.27a) and (2.27b). The improvement is illustrated in the case of fBm(0.3) in Figure 2.3(c) which displays the realized covariance matrix associated with the new MAR-Haar model. This shows the power of the optimal prediction procedure in the simple case of the Haar wavelet.

Despite the improvement in modeling fBm(0.3) obtained by MAR-Haar dynamics that perform optimal prediction of a child state from its parent, there is considerable room for additional improvement. That is, Figure 2.3(c) is still a relatively poor approximation of Figure 2.3(a). In Chapter 5 we will extend the ideas presented in this section and show how to build MAR models based on *any* orthogonal or biorthogonal wavelet. As we illustrate, statistical fidelity improves as we consider more regular wavelets.

■ 2.3.2 MAR Models for One-Dimensional Markov Processes

In this section we discuss another class of conceptually straightforward MAR models—those for one-dimensional, wide-sense, bilateral Markov (a.k.a. reciprocal) processes developed in [127, 129]. These will be applied to the problem of maximum-entropy covariance extension in Chapter 6. A discrete-time process $z(\cdot)$ is a k-th order wide-sense[9] bilateral Markov process if the LLS estimate of $z(n)$ given all other values of the process depends only on the k values on either side of $z(n)$ [53]. More precisely, $z(\cdot)$ is k-th order bilateral Markov if

$$\widehat{\mathrm{E}}[z(n) \mid \{z(i)\}_{i \neq n}] = \widehat{\mathrm{E}}[z(n) \mid \{z(i)\}_{i \in [n-k:n+k]-\{n\}}] \tag{2.29}$$

where $[a:b]$ is the set of integers larger than or equal to a and smaller than or equal to b. Equation (2.29) is equivalent to the statement that, conditioned on the size-k boundaries of an interval J, the values of $z(\cdot)$ inside J and outside J are uncorrelated.

A unilateral Markov process is similar to a bilateral one but it is one-sided. That is, if $z(\cdot)$ is a k-th order unilateral Markov process then the LLS estimate of $z(n)$ given the entire past depends only on the previous k samples. More precisely,

$$\widehat{\mathrm{E}}[z(n) \mid \{z(i)\}_{i<n}] = \widehat{\mathrm{E}}[z(n) \mid \{z(i)\}_{i \in [n-k:n-1]}] \, . \tag{2.30}$$

Equation (2.30) is equivalent to the statement that, conditioned on a length-k interval J, the values of $z(\cdot)$ subsequent to and preceding J are uncorrelated. Every unilateral Markov process is bilateral Markov but the converse is not true [1,53]. We also point out that every k-th order unilateral Markov process can be written as a state-space process with state dimension k as follows. A k-th order unilateral Markov process obeys an autoregression of the form

$$z(n) = a_{n,1}z(n-1) + a_{n,2}z(n-2) + \cdots + a_{n,k}z(n-k) + \mu(n) \tag{2.31}$$

[9] In this thesis we are concerned exclusively with second-order statistics. Therefore, whenever it is not specifically indicated, statistical properties are to be interpreted as wide-sense. Equivalently, there is no loss in generality by assuming Gaussianity.

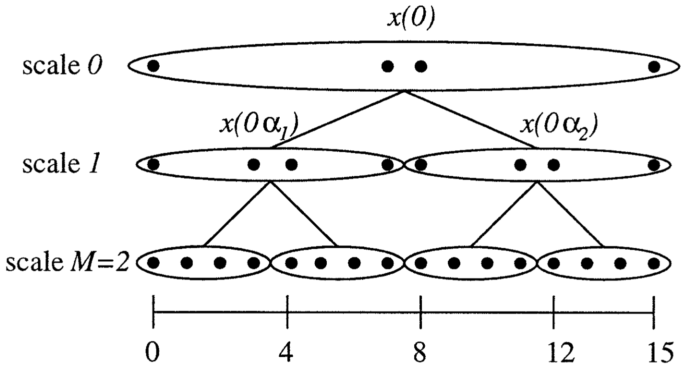


Figure 2.4. MAR model for sixteen samples of a first-order Markov process. Each ellipse includes the samples for a MAR state.

where $\mu(\cdot)$ is white. By defining

$$x(n) \triangleq \left[z(n) \quad z(n-1) \quad \cdots \quad z(n-k+1)\right]^T, \tag{2.32a}$$

$$w(n) \triangleq \left[\mu(n) \quad 0 \quad \cdots \quad 0\right]^T, \tag{2.32b}$$

$$A(n) \triangleq \begin{pmatrix} a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,k-1} & a_{n,k} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{2.32c}$$

we arrive at the state-space representation

$$x(n) = A(n)x(n-1) + w(n). \tag{2.33}$$

The MAR models for Markov processes discussed in [127, 129] are all based on a "divide-and-conquer" philosophy. They qualitatively resemble nested dissection [57, 61, 82, 187] (particularly so the MAR models for two-dimensional MRFs) and quantitatively resemble midpoint displacement for synthesizing Brownian motion [9, 37]. To begin, let us consider just one particular type of MAR model for a length $N = 16$, one-dimensional, first-order Markov processes $f^M(\cdot)$. The MAR model depicted in Figure 2.4 is an *end-point* model for this Markov process. Each state consists of four end-points corresponding to two intervals. For instance,

$$x(0) \triangleq \left[f^M(0) \quad f^M(7) \quad f^M(8) \quad f^M(15)\right]^T, \tag{2.34a}$$

$$x(0\alpha_1) \triangleq \left[f^M(0) \quad f^M(3) \quad f^M(4) \quad f^M(7)\right]^T, \tag{2.34b}$$

$$x(0\alpha_2) \triangleq \left[f^M(8) \quad f^M(11) \quad f^M(12) \quad f^M(15)\right]^T, \tag{2.34c}$$

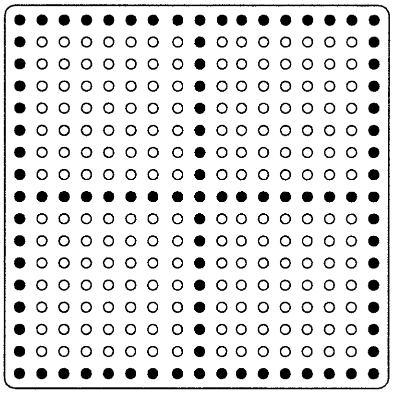


Figure 2.5. The elements of a 17 × 17 MRF contained at the root node of a quad-tree MAR model are indicated by filled circles (●).

and the other states are similarly defined as indicated in Figure 2.4. Notice that $x(0)$ consists of end-points of the intervals $[0:7]$ and $[8:15]$. Therefore, using (2.29), conditioned on $x(0)$, the sets of samples $\{f^M(i)\}_{i\in[0:7]}$ and $\{f^M(i)\}_{i\in[8:15]}$ are uncorrelated. It follows that $x(0\alpha_1)$ and $x(0\alpha_2)$ are uncorrelated when conditioned on $x(0)$. Hence, the statistical relationships among $x(0)$, $x(0\alpha_1)$, and $x(0\alpha_2)$ can be captured *exactly* by a MAR process. Moreover, since the MAR states are linear functionals of x^M, given the fine-scale statistics, P_{f^M}, for any Markov process, the MAR parameters are readily computed (cf., (2.19) and (2.21)). A similar argument applies to the other states.

The MAR model just discussed can be generalized in a number of ways. Most obviously, it can be extended to any length-N process where $N = 4 \times 2^M$ for some integer M. It can also be extended to any order (k) Markov process. Roughly speaking, one way to adapt the preceding approach to a general k is simply to replace every sample (dot, ●) in Figure 2.4 with k consecutive samples. To be more specific, a state s in an $(M+1)$-scale MAR model for a k-th order Markov process consists of the samples of the process indexed by $\eta(s) = \eta_1(s) \cup \eta_2(s) \cup \eta_3(s)$ where

$$\eta_1(s) = \imath(s)4k2^{M-m(s)} + [0:k-1], \tag{2.35a}$$

$$\eta_2(s) = \imath(s)4k2^{M-m(s)} + 4k2^{M-m(s)-1} + [-k:k-1], \tag{2.35b}$$

$$\eta_3(s) = \imath(s)4k2^{M-m(s)} + 4k2^{M-m(s)} + [-k:-1] \tag{2.35c}$$

where $[a:b]+c \triangleq [a+c:b+c]$. Notice that $\eta(s) \subset \eta(s\alpha_1) \cup \eta(s\alpha_2)$ corresponding to the fact that each parent state consists of a subset of the samples in its children states. This approach can accommodate any k-th order Markov processes of length $N = 4k2^M$.

Additional flexibility (i.e., the ability to handle processes of lengths other than $N = 4k2^M$) is accomplished by considering redundant (or overlapping) states and trees that are not binary. The essential characteristics of these other models for Markov processes are identical to the one we have discussed in detail. Since, in this thesis, we have no particular need for the added flexibility they provide, we refer the reader to [37,127,129] for details. For completeness, we point out that in [90] yet another type of model for Markov processes is discussed. For these models, each fine-scale sample appears as an element of only one state, in contrast to the model we have discussed in which a sample may appear in multiple states.

Finally, the Markov models we have discussed may be generalized to apply to wide-sense, two-dimensional MRFs. The essential difference is that MAR states contain lines that separate regions rather than end-points that separate intervals. For instance, the elements contained in the root node of one type of MAR-MRF model are illustrated in Figure 2.5. Notice that a consequence is that MAR states for an MRF model have a maximum dimension proportional to the linear size of the MRF. That is, in modeling a $\sqrt{N} \times \sqrt{N}$ field, the maximum state dimension is proportional to $\sqrt{N}$. This means that the MAR estimator, when applied to such a model, has complexity $O(N^{3/2})$ [160] which is prohibitive for large N. Additional details regarding MAR-MRF models may be found in [37,90,127,129].

■ 2.3.3 Canonical Correlations Realization

The MAR realization approach discussed in the previous section is tailored specifically for Markov processes. In this section we summarize a more general realization approach based on *canonical correlations* and developed in [96,98,100]. This approach has been successfully applied to Markov as well as non-Markov processes. Moreover, it provides a way to prioritize state information so that the least important information may be discarded if a reduced-order, approximate realization is desired.

As mentioned, the heart of the MAR realization problem is to select the linear functionals $\{L_s\}$. The MAR model parameters are then determined using these and P_{f^M} (cf., (2.19) and (2.21)). The realization procedure proposed in [96,98,100] determines these linear functionals by myopically focusing on the state at each node independently. More specifically, the focus is on the decorrelation role that each state fulfills. Recall that, by the Markov property, each MAR state, $x(s)$, conditionally decorrelates the states indexed by nodes that are separated by s. When considering an internal model—in which every state depends linearly on the finest scale sub-process x^M—this decorrelation role simplifies. In this case, it is sufficient that $x(s)$ conditionally decorrelate the fine-scale sub-processes that are separated by s, namely those in the set $\{x^M_{s\alpha_i}\}_{i=1}^q \cup \{x^M_{s^c}\}$. We will make this precise in Chapter 3. Thus, the problem of determining L_s is identical to that of deducing what information must be stored in $x(s)$ to fulfill this simpler decorrelation role.

The determination of the matrices $\{L_s\}$ is based on canonical correlations analysis which is a technique first introduced in [94] and later employed to find minimal and reduced-order realizations for state-space processes [6–8,49,54,183]. As discussed, for a MAR process each state conditionally decorrelates several (generally more than two) sets of variables. Canonical correlations analysis, however, shows how to find a prioritized ordering of the information required to conditionally decorrelate *two* random vectors, z_1 and z_2 with auto-covariances P_1 and P_2, respectively, and cross-covariance

P_{12}. We first review this technique and then show how it may be extended to deal with more than two random vectors.

We first define the *generalized correlation coefficient* as

$$\bar{\rho}(z_1, z_2) \triangleq \max_{g_1, g_2} \frac{\mathrm{E}\Big[\Big(g_1^T(z_1 - \mathrm{E}[z_1])\Big)\Big(g_2^T(z_2 - \mathrm{E}[z_2])\Big)\Big]}{\operatorname{var}(g_1^T z_1)^{1/2} \operatorname{var}(g_2^T z_2)^{1/2}} \tag{2.36a}$$

$$= \max_{\substack{g_1^T P_1 g_1 = 1 \\ g_2^T P_2 g_2 = 1}} g_1^T P_{12} g_2 \,. \tag{2.36b}$$

Similarly, the *generalized conditional correlation coefficient* is

$$\bar{\rho}(z_1, z_2 \mid y) \triangleq \bar{\rho}(\widetilde{z}_1, \widetilde{z}_2) \,. \tag{2.37}$$

where $\widetilde{z}_i \triangleq z_i - \widehat{\mathrm{E}}[z_i \mid y]$ is the error in the LLS estimate of z_i based on y. Canonical correlations analysis provides a linear function of $z = [z_1^T \; z_2^T]^T$ that conditionally decorrelates z_1 and z_2, i.e., it provides a V to yield $\bar{\rho}(z_1, z_2 \mid Vz) = 0$. Further, it tells us which elements of Vz contain the most (and least) important decorrelating information. The main result from canonical correlations analysis is the following proposition.

Proposition 2.3.1 ([96, 100]). *Suppose $P_i \in \mathbb{R}^{n_i \times n_i}$ has rank m_i for $i = 1, 2$ and $P_{12} \in \mathbb{R}^{n_1 \times n_2}$ has rank m_{12}. Then there exist matrices T_1 and T_2 such that*

$$\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} \begin{bmatrix} P_1 & P_{12} \\ P_{21} & P_2 \end{bmatrix} \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}^T = \begin{bmatrix} I_{m_1} & D \\ D^T & I_{m_2} \end{bmatrix} \tag{2.38a}$$

where I_ℓ is the $\ell \times \ell$ identity matrix, $T_i \in \mathbb{R}^{m_i \times n_i}$ and $D \in \mathbb{R}^{m_1 \times m_2}$ has the form

$$\begin{bmatrix} \widehat{D} & 0 \\ 0 & 0 \end{bmatrix} . \tag{2.38b}$$

The matrix $\widehat{D}$ is a diagonal matrix $\widehat{D} = \operatorname{diag}(d_1, d_2, \ldots, d_{m_{12}})$ with $d_j \in (0, 1]$ and $1 \geq d_1 \geq d_2 \geq \cdots \geq d_{m_{12}} > 0$.

The numbers d_j are correlation coefficients associated with different particular linear combinations of z_1 and z_2 called *canonical variables*. The canonical variables are described by the rows of T_1 and T_2. In essence, the d_j, called *canonical correlation coefficients*, measure how correlated z_1 and z_2 are. The proof of Proposition 2.3.1 given in [96, Appendix A.1] is constructive and explicitly shows how to compute the matrices T_i. While we shall not repeat the proof here, we show how to compute T_i and comment on the computational complexity of doing so. The first step is to find the inverse of any matrix square root of P_i for $i \in \{1, 2\}$. That is, find a matrix $P_i^{-1/2}$ such that $P_i^{-T/2} P_i^{-1/2} = P_i^{-1}$. This requires $O(n_i^3)$ operations. The second step is to compute the singular value decomposition (SVD) [116, 172] of $\widetilde{P}_{12} \triangleq P_1^{-1/2} P_{12} P_2^{-T/2}$. That is,

$$\widetilde{P}_{12} = U_1 S U_2^T \,. \tag{2.39}$$

This requires $O(n_1^3 + n_2^3)$ operations. Finally $T_i = U_i^T P_i^{-1/2}$ which requires $O(n_i^3)$ operations. Thus, the overall complexity of this computation is $O(n_1^3 + n_2^3)$. We will refer to the computation of T_i as a *canonical correlations decomposition.*

The following proposition relies upon Proposition 2.3.1 and shows exactly what information is most and least important in decorrelating z_1 and z_2 subject to a constraint on the number of elements in the vector used in the decorrelation.

Proposition 2.3.2 ([96, 100]). *Let T_i for $i = 1, 2$ and d_j for $j = 1, 2, \ldots, m_{12}$ be as defined in Proposition 2.3.1 and let r be the number of elements in the vector used to approximately decorrelate z_1 and z_2. Let $z = \begin{bmatrix} z_1^T & z_2^T \end{bmatrix}^T$ and define $\mathcal{M}_r$ to be the set of matrices with no more than r rows (and the number of columns given by the context). Then*[10]*, for $i = 1, 2$*

$$\min_{V \in \mathcal{M}_r} \bar{\rho}(z_1, z_2 \mid Vz) = \min_{V \in \mathcal{M}_r} \bar{\rho}(z_1, z_2 \mid Vz_i) \tag{2.40a}$$

$$= \bar{\rho}\left(z_1, z_2 \mid T_i\left(1 : \min(r, m_{12}), :\right) z_i\right) \tag{2.40b}$$

$$= \begin{cases} d_{r+1} & r < m_{12}\,, \\ 0 & \textit{otherwise} \end{cases}. \tag{2.40c}$$

Proposition 2.3.2 has a number of important implications. First, it is sufficient to condition on a linear function of either z_1 or z_2. It is *never* necessary to consider all of z. Second, the information which provides the maximal decorrelation subject to the row constraint is provided by the first $\min(r, m_{12})$ rows of either T_1 or T_2. Finally, one need not compute both T_1 and T_2, one or the other will suffice.

Now we consider the decorrelation of more than two random vectors. First define (with an abuse of notation) the correlation between an arbitrary number of random vectors, $z_1, z_2, \ldots, z_{q+1}$, as

$$\bar{\rho}(z_1, z_2, \ldots, z_{q+1}) \triangleq \max_{i \neq j} \bar{\rho}(z_i, z_j)\,, \tag{2.41}$$

with the obvious natural analogue for the conditional correlation for an arbitrary number of random vectors. The problem is to find the prioritized ordering of the information in $z = \begin{bmatrix} z_1^T & z_2^T & \cdots & z_{q+1}^T \end{bmatrix}^T$ needed to conditionally decorrelate the $q + 1$ sub-vectors in z. Unlike the pair-wise problem discussed previously, there is no known solution to this higher-order decorrelation problem. The suboptimal approach of [96, 98, 100] is to decompose the problem into q pair-wise decorrelation problems identical to the pair-wise problem which is the subject of Proposition 2.3.1 and Proposition 2.3.2. More specifically, the i-th pair-wise decorrelation problem is the one of finding the essential information required to conditionally decorrelate z_i from the vector z_i^c which is a vector containing z_j for $j \neq i$, i.e., $z_i^c \triangleq \begin{bmatrix} z_1^T & z_2^T & \cdots & z_{i-1}^T & z_{i+1}^T & \cdots & z_q^T & z_{q+1}^T \end{bmatrix}^T$. For this purpose, the following proposition is essential.

[10] We use MATLAB [137] notation in (2.40b). That is, for any matrix F, the object $F(r_1 : r_2, :)$ is the submatrix of F consisting of the rows r_1 through r_2 (inclusive) and all of the columns.

Proposition 2.3.3 ([96, 100]). *For $i = 1, 2$ and all matrices V*

$$\bar{\rho}(z_1, z_2 \mid Vz_i) \leq \bar{\rho}(z_1, z_2)\,. \tag{2.42}$$

Proposition 2.3.3 tells us that conditioning on a linear combination of either z_1 or z_2 alone cannot increase the correlation between the two. The linear functionals L_s can therefore be computed as follows. For each $i \in \{1, 2, \ldots, q\}$, solve the following pair-wise decorrelation problem: compute the linear combination of[11] $f^M_{s\alpha_i}$ that conditionally decorrelates it from the $f^M_{s\alpha_i^c}$, the rest of the finest-scale sub-process. This is done via canonical correlations as described previously. Then, by stacking these linear combinations column-wise into a large vector, L_s is defined. Using Proposition 2.3.3, it can be shown that the state so defined conditionally decorrelates the set of vectors $\{f^M_{s\alpha_i}\}_{i=1}^q \cup \{f^M_{s^c}\}$ and, thus, the Markov property is satisfied. The model parameters are then determined using (2.19) and (2.21). A proof that this results in an exact model ($P_{x^M} \equiv P_{f^M}$) is given in [100].

If a reduced-order, approximate model is desired, the least important state information may be discarded. This reduction in state dimension corresponds to discarding certain rows of L_s prior to computing the model parameters. The determination of which rows are least important is made by ranking the canonical correlations coefficients. Of course, if state reduction is performed then the Markov property is no longer satisfied. However, by assuming that it holds, we may still define an approximate MAR model. A consequence is that any discrepancy between the realized covariance, P_{x^M}, and the given one, P_{f^M}, can be attributed to state dimension reduction.

Having summarized the realization approach of [96, 98, 100] we now comment on some limitations.

- The higher-order decorrelation problem that must be solved at each node is done so suboptimally via a sequence of pair-wise problems. This may lead to suboptimal state dimensions, i.e., states with more variables than are necessary to fulfill the decorrelation role to the desired degree of approximation. Moreover, *any* approach based on internality (including those proposed in this thesis) cannot, with certainty, yield a minimal model—one with the smallest possible state dimensions. Indeed, it is explicitly shown in [96] that the class of internal models does not necessarily include a minimal model.

- The approach (as well as those proposed in this thesis) does not minimize any measure of global error and, hence, it is not clear how state reduction translates into an error in the realization.

[11] $f^M_{s\alpha_i}$ is the portion of f^M that is indexed by $\mathcal{T}_{s\alpha_i}(M)$, the finest-scale nodes that descend from $s\alpha_i$. Similarly, $f^M_{s\alpha_i^c}$ is the portion of f^M that is indexed by $\mathcal{T}^c_{s\alpha_i}(M)$, the finest-scale nodes that do not descend from $s\alpha_i$.

- The generalized correlation coefficient is normalized by the variance of both z_1 and z_2 (cf., (2.36)). As such, the canonical correlations approach provides maximal *fractional* error variance reduction and is, therefore, equally concerned with low- and high-variance features. In contrast, the *predictive efficiency* approach developed in this thesis provides maximal *total* error variance reduction.

- The algorithm is computationally intensive. This is due to the fact that the design of L_s at every node s requires the computation of several canonical correlations decompositions. Each one involves two vectors at least one of which has length $O(N)$ where N is the length of the fine-scale process x^M. Thus, the overall complexity of the algorithm is $O(N^4)$. The only way to apply such an algorithm to large problems is to make approximations. Indeed, an approximation is proposed in [96,100] which is based on the idea of truncating the large, length-$O(N)$ vectors. We will elaborate on this idea in Chapter 4. Another type of approximation that is introduced in [37,40] is applicable to self-similar processes with stationary increments and leads to an $O(N^3)$ realization algorithm. In contrast, in Chapter 4, we develop an $O(N^2)$ approach and, with an approximation, an $O(N)$ one.

- The approach of [96, 98, 100] does not provide a way to incorporate specific nonlocal linear functionals of the finest scale process. The ability to do just this is important for data fusion problems involving measurements at multiple resolutions. A method of augmenting the approach of [96, 98, 100] so that specific nonlocal linear functionals may be included is provided in [37,39] and is summarized in Section 2.3.5 of this chapter. In Chapter 7 a different and more powerful approach is described.

- The states are defined abstractly and have no structure beyond the fact that they represent solutions to specific optimization problems. In Chapter 5 we address this issue and develop internal MAR processes with states that consists of wavelet coefficients associated with any orthogonal or biorthogonal wavelet.

- The approach assumes complete and detailed knowledge of P_{f^M} which is an unreasonable assumption for many real-world problems. In Chapter 6 we discuss a means of obtaining realizations without complete second-order characterization of f^M.

- Low-dimensional, approximate models lead to discontinuous artifacts in sample-paths and estimates. This problem can be overcome by increasing the state dimension, but doing so leads to less efficient statistical inference algorithms. However, another approach, the overlapping tree methodology of [63, 96, 97], provides a means of obtaining smooth sample-paths and estimates without increasing the state dimension. This technique will be reviewed in Section 2.3.4.

- Finally, the resulting MAR model is inconsistent and not necessarily internal. This problem is addressed and resolved in Chapter 3.

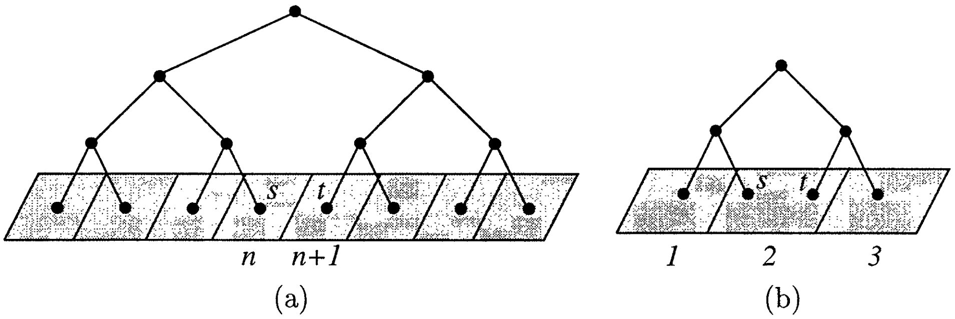


Figure 2.6. (a) Tree nodes s and t correspond to temporal indices n and $n+1$, respectively. While n and $n+1$ are temporal neighbors, s and t are distantly separated on the tree. (b) An overlapping-tree representation of a length-three process. Nodes s and t correspond to the same temporal index, 2.

While some of these limitations are addressed in this thesis, others represent open problems for future research and are discussed in greater detail in Chapter 8.

■ 2.3.4 Overlapping Trees

In this section we summarize the overlapping tree approach of [63, 96, 97]. Overlapping trees represent a way to overcome the distracting and, in some cases, practically limiting problem of blockiness exhibited by sample-paths and estimates based on low-dimensional MAR models (for an example, see Figure 4.9). While the intricacies of implementation of overlapping trees are important they will not be found here as they are clearly presented in [96,97] and in even greater detail in [63]. For the purposes of this thesis, it is sufficient to sketch the main ideas which are conceptually simple and practically powerful. We provide this sketch for one-dimensional problems corresponding to MAR models on dyadic trees. Extensions of the concepts we discuss to two-dimensional problems corresponding to MAR models on quad-trees are straightforward and details are found in [63, 96, 97].

In some cases, the visually distracting blockiness exhibited by low-dimensional MAR models for images is a practical limitation. One such case is when gradients must be taken [63, 69]. While blockiness can be eliminated by post-processing sample-paths and estimates with a low-pass filter, doing so will eliminate not only blocky artifacts but also statistically meaningful fine-scale details. Moreover, such post-processing will necessitate additional and, possibly, computationally costly processing to compute the estimation error statistics. The overlapping tree approach provides an alternative method that retains many of the attractive features of the MAR framework (e.g., efficient computation of both estimates *and* error variances) and reduces blocky artifacts *without* spatial averaging.

The source of blocky artifacts stems from the topology of trees. Consider a reduced-dimension (i.e., approximate) MAR model indexed by the tree in Figure 2.6(a). The fine-scale nodes labeled s and t correspond to temporal indices n and $n+1$, respectively. While n and $n+1$ are neighbors temporally, s and t are distant on the tree. This distant separation is informally called a *tree boundary* although this terminology has no precise meaning. The modeled correlation between variables indexed by s and t, namely, $x(s)$ and $x(t)$, is a function of the statistics at all nodes on the path from s to t (cf., (2.17)). Therefore, the modeled correlation suffers from approximations made at all nodes on the path between s and t, of which there are many. So, while the *exact* correlation between $x(s)$ and $x(t)$ may be high (as might be expected since they correspond to temporally adjacent variables), this high correlation is poorly approximated by a low-dimensional model. This results in a discontinuity in the model's statistics and leads to blockiness.

The overlapping tree approach to eliminating blockiness is to oversample the process being modeled to obtain a redundant one. The redundancy is designed specifically so that a MAR model for the redundant process has multiple leaf nodes corresponding to process values that would otherwise be distantly separated on the tree (see Figure 2.6(b)). Consequently, the tree distance between nodes is significantly reduced as compared to a non-redundant model. After building a MAR model for the redundant process, which we may do using *any* realization approach, we may use it to accomplish sample-path generation and estimation. Then, the sample-paths and estimates may be mapped back into the original (non-redundant) domain by simply averaging leaf-node variables corresponding to each *individual* pixel. Thus, this averaging of MAR variables does *not* introduce spatial averaging in the image domain. As demonstrated in [63, 96, 97], to achieve the same degree of smoothness, overlapping MAR models typically require substantially *lower* state dimensions as compared to standard MAR models.

As a specific example, consider Figure 2.6(b) which illustrates an overlapping MAR model indexed on a three-scale dyadic tree where the process to be modeled, f^M, is length-three. At the finest scale there are two nodes corresponding to temporal index 2 while there is just one node for temporal index 1 and one node for temporal index 3. That is, in our redundant representation of f^M, we have sampled $f^M(2)$ twice and associated each copy with a *different* finest-scale MAR state. More precisely, the finest-scale redundant process is Gf^M where

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.43}$$

To map back to the original (non-redundant) domain without spatial averaging, we simply need to average the two variables indexed by nodes s and t. Thus, we are lead to a left-inverse of G given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a & b & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.44}$$

where $a + b = 1$ and $HG = I_3$. For example, the choice of $a = b = 1/2$ would place an equal weight on $x(s)$ and $x(t)$ and, intuitively, would lead to the greatest degree of smoothness.

The concepts highlighted in the foregoing example generalize to any one-dimensional processes as well as to two-dimensional ones. Moreover, as described in [63,96,97], there is a complementary procedure for mapping fine-scale, point-wise measurements into the overlapped domain. It is worth emphasizing two limitations of overlapping trees. First, it is unclear how to accommodate nonlocal measurements, such as those that arise in multiresolution data fusion problems, using the overlapping approach. Second, while overlapping reduces statistically meaningless blocky artifacts by spreading the modeling error more evenly, by itself it *cannot* substantially increase the overall statistical fidelity of the model. The latter may only be done by increasing the state dimension.

■ 2.3.5 Adding Nonlocal Variables by State Augmentation

To date, most of the applications of the MAR framework focus on the statistical inference of fine-scale phenomena based on fine-scale data. Therefore, in these applications, coarser-scale MAR variables serve the hidden variable role of conditional decorrelation and do not necessarily represent linear functionals of the finest scale that are useful for any other purpose. However, data fusion problems in which data are available at multiple resolutions are most naturally addressed in the MAR framework by mapping the coarser-scale data to coarser-scale nodes [37, 39]. In addition, in many applications, estimates of specific coarser-scale variables are of interest and these too are most naturally mapped to coarser-scale MAR variables [37,41]. The problem of incorporating specific nonlocal linear functionals (representing either nonlocal measurements or nonlocal variables to be estimated) of a fine-scale process into a MAR model is not addressed by the realization approaches we have discussed so far. In this section we discuss the method developed in [37–39] to address this issue.

The approach of [37–39] begins with an existing internal MAR model which is assumed to capture the desired fine-scale statistics with sufficient accuracy to be considered exact. That is, the $\{L_s\}$ have already been determined. Since the model is, for all practical purposes, exact and internal, x^M is sufficiently close to f^M so that MAR states may be considered linear functions of the fine-scale process f^M rather than of x^M. So, $x(s) = L_s f^M$ and, thus, the MAR measurement equation (2.18) has the form

$$y(s) = C(s)x(s) + v(s) \tag{2.45a}$$

$$= C(s)L_s f^M + v(s)\,. \tag{2.45b}$$

Therefore, to represent a nonlocal measurement of the form $y = a^T f^M + \nu$ exactly requires that for some $s \in \mathcal{S}_0$, a^T is in the row-space of L_s. Similarly, if we wish to use the MAR estimator to estimate and compute the estimation error variance for $b^T f^M$, we require that b^T is in the row-space of some L_s. Having already chosen the linear functionals $\{L_s\}$ in designing an exact, internal MAR model for f^M, it is unlikely that an arbitrary a^T or b^T is in the row-space of any of them. However, by augmenting MAR states to expand the set of linear functionals they can represent, the authors of [37–39] incorporate nonlocal variables in such a way so as to maintain an exact and internal model.

To maintain the exactness of the model corresponds to preserving its Markovianity. To see how Markovianity can be destroyed with state augmentation, consider a MAR end-point model for a length-16, first-order Markov process as described in Section 2.3.2 and illustrated in Figure 2.4. Suppose we wish to represent at the root node the sum of the fourth and twelfth samples of f^M, samples which are conditionally decorrelated by the root node state. To do so, we might consider augmenting the root-node state with a linear functional a^T such that $a^T f^M = f^M(4) + f^M(12)$. That is, consider redefining the root-node state to be

$$x_{\text{new}}(0) = \begin{bmatrix} L_0 \\ a^T \end{bmatrix} f^M \tag{2.46}$$

where L_0 is as indicated in Figure 2.4 and (2.34a). However, by incorporating this sum into $x_{\text{new}}(0)$, $f^M(4)$ (which is an element of $x(0\alpha_1)$) and $f^M(12)$ (which is an element of $x(0\alpha_2)$) are *not* conditionally decorrelated by $x_{\text{new}}(0)$. The incorporation of their sum into the root node has destroyed the Markovianity of the model and, hence, has destroyed its exactness.

To maintain Markovianity, the authors of [37–39] rely on a corollary of Proposition 2.3.3. This corollary states that we will not destroy Markovianity by adding to any state $x(s)$ (of an exact MAR model) linear functions of the portions of f^M indexed by finest-scale nodes that are separated by s. That is, we may add linear functions of the form $a^T f^M_{s\alpha_i}$ or $b^T f^M_{s^c}$. Therefore, if we wish to add the linear function $g^T f^M$ to the state at node s without destroying Markovianity, we may do so as follows. Let

$$G_s = \begin{pmatrix} g^T_{s\alpha_1} & 0 & \cdots & 0 & 0 \\ 0 & g^T_{s\alpha_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & g^T_{s\alpha_q} & 0 \\ 0 & 0 & \cdots & 0 & g^T_{s^c} \end{pmatrix} \tag{2.47}$$

where

$$g^T f^M = \sum_{i=1}^{q} g^T_{s\alpha_i} f^M_{s\alpha_i} + g^T_{s^c} f^M_{s^c} \,. \tag{2.48}$$

With these definitions, we may augment $x(s)$ to form

$$x_{\text{new}}(s) \triangleq \begin{bmatrix} L_s \\ G_s \end{bmatrix} f^M \tag{2.49}$$

without perturbing the Markovianity of the model. If none of the row-space of G_s is contained in the row-space of L_s then this increases the dimension of the state at node s from d_s to $d_s + q + 1$. If, on the other hand, part of the row-space of G_s is in the row-space of L_s, then the dimensionality of the augmentation may be reduced by a commensurate amount.

While the foregoing procedure does maintain Markovianity, it will, in general, destroy internality. Internality is destroyed if the information added to $x(s)$ through augmentation is not propagated down the tree to the finest scale. Therefore, some additional state augmentation is required to achieve this propagation of information. We illustrate the general procedure given in [37,39] to accomplish this with an example.

Consider again the end-point model for a length-16 Markov process as discussed in Section 2.3.2 and illustrated in Figure 2.4. Suppose we wish to augment the root node with the sum, h, of the entire finest-scale sub-process. As described above, we can do so without destroying Markovianity by augmenting the root-node state with the linear functions

$$h_1 = \sum_{k=0}^{7} f^M(k) \quad \text{and} \quad h_2 = \sum_{k=8}^{15} f^M(k)\,. \tag{2.50}$$

So, $h = h_1 + h_2$ and $x_{\text{new}}(0) = \begin{bmatrix} x(0)^T & h_1 & h_2 \end{bmatrix}^T$. Now, to ensure that the sum h captured at the root node is indeed equal to the sum of fine-scale variables, this information must be propagated from the root node to the finest scale. One way to accomplish this with augmentation while simultaneously maintaining Markovianity is with

$$x_{\text{new}}(0\alpha_1) \triangleq \begin{bmatrix} x(0\alpha_1) \\ \sum_{k=0}^{3} f^M(k) \\ \sum_{k=4}^{7} f^M(k) \end{bmatrix} \quad \text{and} \quad x_{\text{new}}(0\alpha_2) \triangleq \begin{bmatrix} x(0\alpha_2) \\ \sum_{k=8}^{11} f^M(k) \\ \sum_{k=12}^{15} f^M(k) \end{bmatrix}\,. \tag{2.51}$$

Notice that $x_{\text{new}}(0\alpha_1)$ contains the information provided in h_1, and $x_{\text{new}}(0\alpha_2)$ contains the information provided in h_2. This completes the augmentation required for this example. To obtain the new dynamical parameters, we may apply (2.19) and (2.21) where we augment L_s with the newly added linear functionals.

The preceding example illustrates the important aspects of the general approach detailed in [37,39] for adding a linear functional to any state of an internal and exact MAR model. The aspects of this procedure that are important for this thesis are as follows.

- Adding a linear functional requires the augmentation of many states, not just the one at which the linear functional is represented.

- The procedure may accommodate any number of linear functionals by iterating the operations required to add just one.

- The procedure begins with an exact, internal MAR model and does not address the problem of building a reduced-order and internal MAR model with nonlocal variables. While in [37] a procedure for constructing a reduced-order and internal MAR model with nonlocal variables is provided, its computational complexity severely limits its practical utility.

- For each linear functional added, the maximum state dimension increases by an amount that is a function of the support of the linear functional and the space spanned by the rows of $\{L_s\}$. The increase may be as large as $q+1$. This leads to models with prohibitive complexity if the number of linear functionals added is large.

- The resulting states may be singular, implying that the information they carry is redundant. Similarly, augmented states *always* contain more information than is necessary for Markovianity. Indeed, the information carried by the nonlocal linear functionals is *not* needed for conditional decorrelation since it is assumed that the model is exact prior to the incorporation of nonlocal variables.

In Chapter 7 several other approaches to incorporating nonlocal variables into a MAR model are introduced which address the last three of these points.

2.4 Predictive Efficiency

As discussed in Section 2.3.3, at the heart of MAR stochastic realization is the problem of conditionally decorrelating random vectors. Canonical correlations is one way to approach this problem but it scales poorly with problem size and leads to computationally burdensome realization algorithms. In this section we present a different approach which is based on the estimation-theoretic concept of predictive efficiency[12] [8, 154] and which is *much* more attractive from a computational point of view. In Chapter 4 and Chapter 7, predictive efficiency will be used to address the problem of stochastic realization. To begin, we define $\varepsilon(z_2 \mid z_1)$ to be the mean-square error in the LLS estimate of the length-n_2 vector z_2 based on the length-n_1 vector z_1:

$$\varepsilon(z_2 \mid z_1) \triangleq \mathrm{E}\Big(\big\|z_2 - \widehat{\mathrm{E}}[z_2 \mid z_1]\big\|^2\Big) \tag{2.52a}$$

$$= \operatorname{trace}\big(P_2 - P_{12}^T P_1^{-1} P_{12}\big) \tag{2.52b}$$

where P_i is the positive-definite covariance matrix for z_i, and P_{12} is the cross-covariance matrix for z_1, z_2.

Consider now the problem of estimating z_2 not from z_1 but from no more than r linear functionals of z_1 given by Vz_1 where $V \in \mathcal{M}_r$ which is the set of all matrices of size $\ell \times n_1$ with $\ell \leq r$. We can measure the quality of the estimate based on Vz_1

[12] Most of the material of this section can be found in [77].

relative to that which can be obtained from z_1 by

$$\begin{aligned}\bar{\varepsilon}(z_2 \mid Vz_1) &\triangleq \varepsilon(z_2 \mid Vz_1) - \varepsilon(z_2 \mid z_1) &\text{(2.53a)}\\ &= \operatorname{trace}\bigl(P_{12}^T P_1^{-1} P_{12}\bigr) - \operatorname{trace}\bigl(P_{12}^T V^T (V P_1 V^T)^{-1} V P_{12}\bigr)\,. &\text{(2.53b)}\end{aligned}$$

The idea of predictive efficiency is to minimize $\bar{\varepsilon}(z_2 \mid Vz_1)$ over $\mathcal{M}_r$. Let

$$\widehat{V} \triangleq \arg\min_{V \in \mathcal{M}_r} \bar{\varepsilon}(z_2 \mid Vz_1)\,. \tag{2.54}$$

Notice that the minimum is lower bounded by zero and equality obtains if and only if Vz_1 conditionally decorrelates z_1 and z_2. Therefore, we can interpret $\bar{\varepsilon}(\cdot \mid \cdot\cdot)$ as a measure of distance from Markovianity although it is not a true distance because it is not symmetric. The optimal $V \in \mathcal{M}_r$ according to the predictive efficiency measure is provided in the following proposition.

Proposition 2.4.1. *Let $U\Lambda U^T$ be the eigen-decomposition of $P_1^{-1/2} P_{12} P_{12}^T P_1^{-T/2}$ with the eigenvalue matrix $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_1})$ and $\lambda_i \geq \lambda_j$ for $i \leq j$. Let $r \leq n_1$. Then $\widehat{V} \triangleq \arg\min_{V \in \mathcal{M}_r} \bar{\varepsilon}(z_2 \mid Vz_1)$ is given by the first r rows of $U^T P_1^{-1/2}$ and*

$$\bar{\varepsilon}(z_2 \mid \widehat{V} z_1) = \sum_{i=r+1}^{n_1} \lambda_i\,. \tag{2.55}$$

Proof. By definition of $\bar{\varepsilon}(\cdot \mid \cdot\cdot)$ we have that

$$\bar{\varepsilon}(z_2 \mid Vz_1) = \operatorname{trace}\bigl(P_{12}^T P_1^{-1} P_{12}\bigr) - \operatorname{trace}\bigl(P_{12}^T V^T (V P_1 V^T)^{-1} V P_{12}\bigr)\,. \tag{2.56}$$

Therefore, we must show that

$$\max_{V \in \mathcal{M}_r} \left\{\operatorname{trace}\left(P_{12}^T V^T (V P_1 V^T)^{-1} V P_{12}\right)\right\} = \sum_{i=1}^{r} \lambda_i \tag{2.57}$$

and that the maximum is achieved when V is the first r rows of $U^T P_1^{-1/2}$. We may assume without loss of generality that $V P_1 V^T = I$, i.e., that the transformation of z_1 yields uncorrelated random variables. Indeed, if it did not we could use Gram-Schmidt orthogonalization to find an equivalent set of uncorrelated random variables. Thus the problem is reduced to the one of considering

$$\max_{\{v_i\}} \sum_{i=1}^{r} v_i^T P_{12} P_{12}^T v_i \tag{2.58}$$

where v_i^T is the i-th row of V.

As shown in [149, 154], the maximum of an expression of the form (2.58) with the constraint that $V P_1 V^T = I$ is $\sum_{i=1}^{r} \beta_i$ where $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_{n_1} \geq 0$ and β_i is the i-th eigenvalue associated with the symmetric-definite generalized eigen-problem

$$\det\Bigl(P_{12} P_{12}^T - \beta P_1\Bigr) = 0\,. \tag{2.59}$$

Also the optimal $\{v_i\}$ are the associated generalized eigenvectors. Assuming, as we have throughout, that $P_1 > 0$ we can rewrite this eigen-problem as

$$\det\left(P_1^{-1/2} P_{12} P_{12}^T P_1^{-T/2} - \beta I\right) = 0 \tag{2.60}$$

where the roots (the eigenvalues) of (2.60) are the same as those of (2.59). That is, $\beta_i \equiv \lambda_i$. It is easy to verify that an eigenvector, u, associated with the problem (2.60) is related to an eigenvector, v, associated with the problem (2.59) by $v = P_1^{-T/2} u$ or $v^T = u^T P_1^{-1/2}$. Therefore, the optimal choice of the $\{v_i\}$ is given by $v_i^T = u_i^T P_1^{-1/2}$ for $i = 1, 2, \dots, r$. This completes the proof. ■

In the sequel, we call the pair of matrices (U, Λ) of Proposition 2.4.1 the *predictive efficiency matrices.* The computational complexity of computing these matrices is $O(n_1^2 n_2 + n_1^3)$. The $n_1^2 n_2$ term comes from the formation of the matrix

$$P_1^{-1/2} P_{12} P_{12}^T P_1^{-T/2} . \tag{2.61}$$

The n_1^3 term comes from the fact that we must invert the matrix square root of P_1 and compute an eigen-decomposition of an $n_1 \times n_1$ matrix. The inversion of P_2 is not required because the predictive efficiency method is asymmetric. In fact, P_2 plays no role in the computation of the predictive efficiency matrices. In contrast, canonical correlations requires the inversion of *both* P_1 and P_2 because it is symmetric. It is precisely this difference in symmetry that accounts for the efficiency of the realization algorithms developed in this thesis as compared to those based on canonical correlations. As we will see, in the context of the MAR stochastic realization problem, n_2 is related to problem size, N, while n_1 is related to state dimension and can be chosen to be independent of n_2. Thus, the asymptotic complexity of the predictive efficiency method is $O(N)$ whereas that of the canonical correlations approach is $O(N^3)$.

The main message of Proposition 2.4.1 is that $\widehat{V} z_1$ does the best job (in the sense of (2.54)) of conditionally decorrelating z_1 and z_2 subject to the constraint that $\widehat{V}$ may have no more than r rows. We have a need to generalize this idea to consider the problem of (approximately) conditionally decorrelating more than two random vectors. Consider, for instance, conditionally decorrelating the set of random vectors $\{z_i\}_{i=1}^{q+1}$ with a linear function of $z_0 \triangleq \begin{bmatrix} z_1^T & z_2^T & \cdots & z_q^T \end{bmatrix}^T$. For this purpose, we define (with an abuse of notation) the following generalization[13] of $\bar{\varepsilon}(\cdot \mid \cdot\cdot)$:

$$\bar{\varepsilon}(z_1, z_2, \dots, z_{q+1} \mid V z_0) \triangleq \max_i \{\bar{\varepsilon}(z_i \mid V_i^c z_0)\} \tag{2.62}$$

where $V_i^c z_0$ is the sub-vector of $V z_0$ that *does not* include a the contribution from z_i, and where we assume that V is block diagonal. That is, $V = \operatorname{diag}(V_1, V_2, \dots, V_q)$ so

[13] We interpret $\bar{\varepsilon}(z_i \mid V_i^c z_0)$ as $\bar{\varepsilon}(z_i \mid V_i^c z_0) = \varepsilon(z_i \mid V_i^c z_0) - \varepsilon(z_i \mid z_i^c)$ where $z_i^c = \begin{bmatrix} z_1^T & z_2^T & \cdots & z_{i-1}^T & z_{i+1}^T & \cdots & z_q^T \end{bmatrix}^T$.

that

$$Vz_0 = \begin{bmatrix} V_1 z_1 \\ V_2 z_2 \\ \vdots \\ V_{q-1} z_{q-1} \\ V_q z_q \end{bmatrix} \quad \text{and} \quad V_i^c z_0 = \begin{bmatrix} V_1 z_1 \\ \vdots \\ V_{i-1} z_{i-1} \\ V_{i+1} z_{i+1} \\ \vdots \\ V_q z_q \end{bmatrix} . \tag{2.63}$$

The block diagonal structure imposed on V is motivated by Proposition 2.3.3 which shows that conditioning on a linear function of z_k cannot increase its correlation with z_j for $j \neq k$. In Chapter 4 we will provide a more compelling reason, based on internality, to focus on block diagonal matrices. To be sure, one can consider instances in which it makes sense to relax this block diagonal structure and consider full matrices. We will, in fact, do just this in Chapter 5 when we consider MAR models based on wavelets. For now, however, we restrict attention to block diagonal matrices V.

The corresponding predictive efficiency problem is

$$\widehat{V} = \arg \min_{V \in \mathcal{M}_d^*} \bar{\varepsilon}(z_1, z_2, \dots, z_{q+1} \mid Vz_0) \tag{2.64}$$

where $\mathcal{M}_d^*$ is the subset of $\mathcal{M}_d$ which consists of matrices with the block diagonal structure just described. Note that we can view $\bar{\varepsilon}$ as a measure of Markovianity since Vz_0 conditionally decorrelates $\{z_i\}_{i=1}^{q+1}$ if and only if $\bar{\varepsilon}(z_1, z_2, \dots, z_{q+1} \mid Vz_0) = 0$. Unfortunately, unlike the case for a pair-wise predictive efficiency problem (cf., Proposition 2.4.1), there is no known procedure for solving this higher order predictive efficiency problem. However, by considering q pair-wise predictive efficiency problems instead of (2.64), we can obtain a good suboptimal solution.

Rather than attempt to conditionally decorrelate all $q+1$ random vectors in the set $\{z_i\}_{i=1}^{q+1}$ at once, we instead consider each one in turn. That is, for each $i \in [1 : q]$, we seek a linear function of z_i that (approximately) conditionally decorrelates it from the others. Using the predictive efficiency criterion, this becomes formally

$$V_{i,r_i} = \arg \min_{V \in \mathcal{M}_{r_i}} \bar{\varepsilon}(z_i^c \mid Vz_i) \tag{2.65}$$

where the r_i satisfy $\sum_{i=1}^{q} r_i \leq d$ and $z_i^c = [z_1^T, z_2^T, \dots, z_{i-1}^T, z_{i+1}^T, \dots, z_{q+1}^T]^T$, a vector consisting of z_j for $j \neq i$. This pair-wise problem is solved by computing the predictive efficiency matrices (U_i, Λ_i) as explained in Proposition 2.4.1.

Having solved these q pair-wise problems, we concatenate the resulting matrices V_{i,r_i} to form $\bar{V}$, our suboptimal solution to (2.64):

$$\bar{V} \triangleq \mathrm{diag}(V_{1,r_1}, V_{2,r_2}, \dots, V_{q,r_q}) . \tag{2.66}$$

To completely define our suboptimal solution $\bar{V}$, we need to specify exactly how the r_i are chosen. The first step of our approach is to compute all of the q sets of predictive efficiency matrices $\{(U_i, \Lambda_i)\}_{i=1}^q$. Then, we create one ordered list consisting of all of the eigenvalues and select the largest d eigenvalues from our list, thereby determining the number, r_i, of rows taken from each U_i. To be sure, one can consider other ways of specifying the r_i. Some of these are discussed in Chapter 8.

Chapter 3

Internality and Markovianity

WHILE this thesis is primarily concerned with the identification of MAR models, before we turn to stochastic realization, we need to take a step back to consider tree-indexed processes more generally. There are two basic concepts introduced in this chapter.[1] The first of these is internality which we develop in Section 3.1. Internality has both intellectual and practical importance. First, as described in Chapter 1, it is a natural extension of the well-studied time-series concept. Second, internal MAR models have coarse-scale states that include nonlocal linear functions of fine-scale states. This allows for efficient fusion of nonlocal and local measurements with no increase in computational complexity as compared to the case of fusing only fine-scale data [37,39]. Lastly, while non-internal MAR processes can be constructed, their states have exogenous random components, a property that is not suitable in many problems such as the fusion of multiresolution data. For an internal model, all the statistical properties can be derived from the signal being modeled—there is no exogenous randomness.

The second basic concept associated with tree-indexed processes is Markovianity. As discussed in Chapter 1, we develop a scale-recursive formulation of this concept for internal processes (Section 3.2) that leads to efficient model realization. Once we develop these two basic concepts for tree-indexed processes, we apply them to the MAR stochastic realization problem in Chapter 4. In doing so, we deduce the structure of internal MAR models that both must be satisfied (and which is *not* satisfied by previous methods) *and* which reduces computational complexity.

■ 3.1 Internality

In this section, we first define internality for an arbitrary tree-indexed process and then seek to understand what structure must be imposed on the states of a MAR process to make it internal.

Definition 3.1.1 (Internal Tree-Indexed Process). *A tree-indexed process $x(\cdot)$ is* internal *if for all $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$, $x(s)$ is a linear function of x_s^M, the process indexed by finest-scale nodes that descend from s. That is, for some set of matrices $\{W_s\}$,*

$$x(s) = W_s x_s^M . \tag{3.1}$$

[1] Most of the material of this chapter may be found in [77].

The matrices $\{W_s\}$ are called *internal matrices.* If $x(\cdot)$ is also a MAR process then $x(s)$ defined by (3.1) is called an *internal state.* For a general tree-indexed process, internality places no restrictions on the internal matrices. However, we are interested in internal MAR processes which also obey the MAR dynamics of (2.14). A consequence is that the internal matrices cannot be chosen independently. This can be seen intuitively because fine-scale states are derived from coarse scale ones by (2.14) and must be consistent with the information contained in coarse scale states (given by (3.1)). That is, (2.14) and (3.1) together place constraints on finer-scale states and, thereby, on the internal matrices associated with those states.

The constraints imposed by (3.1) are *not* enforced in previous systematic realization approaches [40, 96, 98, 100]. As a consequence the MAR models developed in these works are *not* internal and their states are not consistent with one another. This fact is illustrated with a specific example in [96, pages 98–99]. Although the authors of [40, 96, 98, 100] were aware of this inconsistency issue, it has, until now, not been dealt with in a theoretically consistent and complete framework. We not only develop a framework that incorporates the constraints of (3.1) but we show how doing so vastly *simplifies* the construction of internal MAR models. The key is that (3.1) is not the right parameterization for internal MAR states. The right parameterization comes from the following.

Definition 3.1.2 (Locally Internal Tree-Indexed Process). *A tree-indexed process $x(\cdot)$ is* locally internal *if for all $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$, $x(s)$ is a linear function of $x_s^{m(s)+1}$, the process indexed by the children nodes of s. That is, for some set of matrices $\{V_s\}$,*

$$x(s) = V_s x_s^{m(s)+1} . \tag{3.2}$$

Notice that (3.1) places all of the focus on the fine scale while (3.2) is scale-recursive. We call V_s a *local internal matrix.* The following proposition shows that the local internal matrices provide the right parameterization for an internal MAR process.

Proposition 3.1.1. *A MAR process, $x(\cdot)$, is internal if and only if it is locally internal.*

Proof. The "if" direction is trivial. If $x(\cdot)$ is locally internal then we may write $x(s)$ as a linear combination of its children. In turn, each $x(s\alpha_i)$ is a linear combination of *its* children and so on, scale-recursively down the tree. Therefore, $x(s)$ is a linear combination of its finest-scale descendents x_s^M. This holds for all $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$ so $x(\cdot)$ is internal.

For the "only if" direction, assume that $x(\cdot)$ is internal. For any $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$, we may write

$$x(s) = \widehat{\mathrm{E}}\left[x(s) \mid x_s^{m(s)+1}\right] + \widetilde{x}(s) \tag{3.3}$$

where $\widetilde{x}(s)$ is uncorrelated with $x_s^{m(s)+1}$. Since $x(s)$ and the $x(s\alpha_i)$ which comprise $x_s^{m(s)+1}$ are assumed to be internal states, $\widetilde{x}(s)$ must be a linear function of x_s^M. We show that $\widetilde{x}(s)$ must be zero (so that $x(s)$ is indeed a linear function of $x_s^{m(s)+1}$). By (3.3), $\widetilde{x}(s)$ is a linear combination of $x(s)$ and $x_s^{m(s)+1}$ so we may write

$$\widetilde{x}(s) = \widehat{\mathrm{E}}\left[\widetilde{x}(s) \mid x(s), x_s^{m(s)+1}\right] = \widehat{\mathrm{E}}\left[\widetilde{x}(s) \mid x_s^{m(s)+1}\right] = 0 \tag{3.4}$$

where the second equality follows from the fact that $\widetilde{x}(s)$ must be a linear function of x_s^M and the fact that conditioned on $x_s^{m(s)+1}$, $x(s)$ and x_s^M are conditionally uncorrelated (by the Markov property). The third equality follows from the fact that $\widetilde{x}(s)$ is uncorrelated with the $x(s\alpha_i)$. This completes the proof. ■

Note that, given the local internal matrices $\{V_s\}$ it is easy to derive the internal matrices $\{W_s\}$ recursively as follows:

$$W_s = \begin{cases} I_d & \text{if } m(s) = M\,, \\ V_s \operatorname{diag}(W_{s\alpha_1}, W_{s\alpha_2}, \ldots, W_{s\alpha_q}) & \text{otherwise} \end{cases} \tag{3.5}$$

where $d \triangleq \dim(x(s))$ is the dimension[2] of the state vector $x(s)$. Since an internal MAR process has states satisfying (3.2) as well as (2.14), we immediately have the following complete characterization of the parameters $A(s)$ and $Q(s)$ for such a process in terms of the local internal matrices and the covariance matrix for $x_s^{m(s)+1}$.

Proposition 3.1.2. *Suppose $x(\cdot)$ is a MAR process. Let $J_{s\alpha_i}$ be the selection matrix such that $J_{s\alpha_i} x_s^{m(s)+1} = x(s\alpha_i)$. Then $x(\cdot)$ is a locally internal MAR process with $x(s) = V_s x_s^{m(s)+1}$ if and only if for all $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$,*

$$A(s\alpha_i) = J_{s\alpha_i} P_{x_s^{m(s)+1}} V_s^T (V_s P_{x_s^{m(s)+1}} V_s^T)^{-1}\,, \tag{3.6a}$$

$$Q(s\alpha_i) = J_{s\alpha_i} \left(P_{x_s^{m(s)+1}} - P_{x_s^{m(s)+1}} V_s^T (V_s P_{x_s^{m(s)+1}} V_s^T)^{-1} V_s P_{x_s^{m(s)+1}} \right) J_{s\alpha_i}^T\,. \tag{3.6b}$$

Proof. See Appendix A. ■

The relations of (3.6) may be written in another form (cf., (2.19)) which emphasizes that $A(s\alpha_i)$ and $Q(s\alpha_i)$ depend only on state covariances and parent-child cross-covariances:

$$A(s\alpha_i) = P_{x(s\alpha_i)x(s)} P_{x(s)}^{-1}\,, \tag{3.7a}$$

$$Q(s\alpha_i) = P_{x(s\alpha_i)} - P_{x(s\alpha_i)x(s)} P_{x(s)}^{-1} P_{x(s\alpha_i)x(s)}^T\,. \tag{3.7b}$$

Together, Propositions 3.1.1 and 3.1.2 provide the necessary and sufficient conditions for a MAR process to be internal.

[2] For clarity of presentation, we restrict attention to processes with constant state dimension, d. However, our results are applicable and easily generalizable to cases in which the state varies with $s \in \mathcal{S}_0$

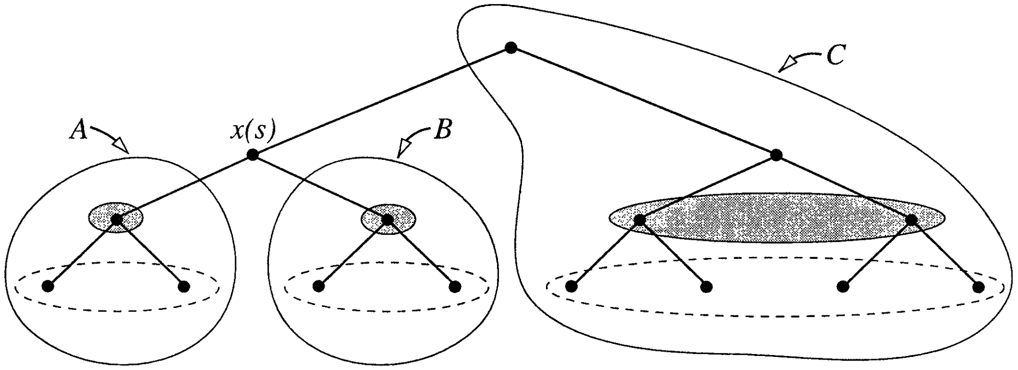


Figure 3.1. For an internal process $x(\cdot)$ indexed by a dyadic tree, the following are equivalent: $x(s)$ conditionally decorrelates the sub-processes indexed by the three sets of nodes (1) in the solid-lined regions labeled "A", "B", and "C" (Markov property), (2) in the dashed-lined regions (fine-scale Markov property), (3) in the shaded regions (scale-recursive Markov property).

■ 3.2 Notions of Markovianity

For internal tree-indexed processes, the Markov property of Definition 2.2.1 is equivalent to two other notions of Markovianity. These notions, which we develop in this section, are much simpler to work with and lead to a scale-recursive realization algorithm, presented in Chapter 4, with substantially reduced computational complexity as compared to previous methods. The first alternate notion of Markovianity is the *fine-scale Markov property*, which is the focus of the approaches in [40, 96, 98, 100].

Definition 3.2.1 (Fine-Scale Markov Property). *A tree-indexed process $x(\cdot)$ has the* fine-scale Markov property *if conditioned on $x(s)$ for any $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$ the $q+1$ vectors in the set $\{x^M_{s\alpha_i}\}_{i=1}^q \cup \{x^M_{s^c}\}$ are conditionally uncorrelated.*

Figure 3.1 provides some intuition about the relationship between the Markov property and the fine-scale Markov property. The Markov property focuses on the conditional decorrelation of the states indexed by the nodes in the subtrees extending from s (the three sets of nodes enclosed by solid lines and labeled "A", "B", and "C" in Figure 3.1). The fine-scale Markov property, in contrast, places its attention on the conditional decorrelation of finest-scale sub-processes (dotted-lined regions in Figure 3.1). While there are fewer leaf nodes than nodes in the tree, this does not provide a substantial amount of simplification because the number of tree nodes and the number of leaf nodes are of the same order. This is the key to why previous realization methods [40, 96, 98, 100], which make extensive use of the fine-scale Markov property, scale poorly with problem size. We now show that, for internal processes, the fine-scale Markov property is equivalent to the Markov property.

Proposition 3.2.1. *Assume that $x(\cdot)$ is an internal tree-indexed process. Then it has the fine-scale Markov property if and only if it has the Markov property.*

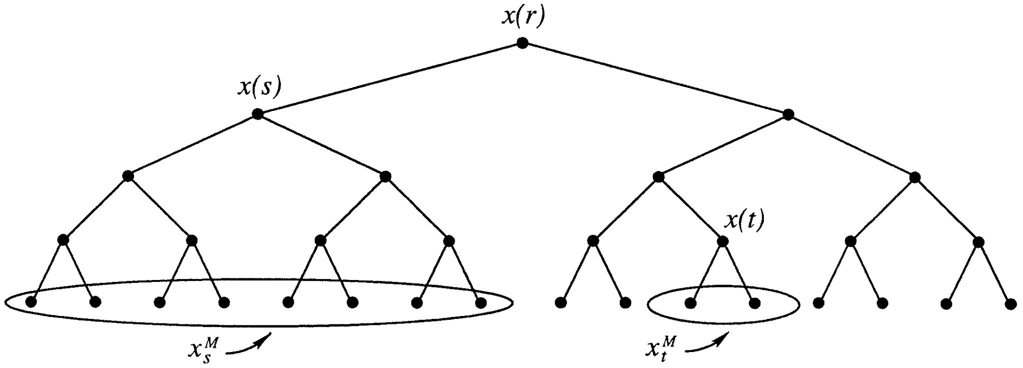


Figure 3.2. Proof of Proposition 3.2.1, case 1: t is not an ancestor of s and s is not an ancestor of t.

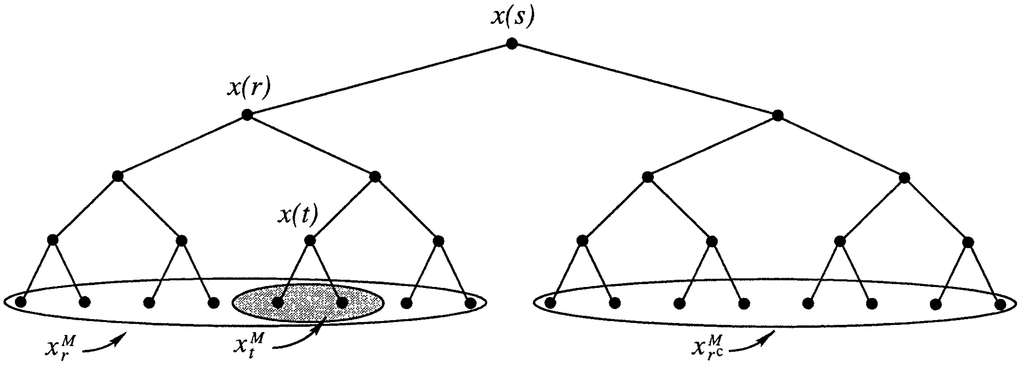


Figure 3.3. Proof of Proposition 3.2.1, case 2: $s = t\bar{\gamma}^n$. The shaded region indicates x_t^M.

Proof. First, if $x(\cdot)$ has the Markov property it clearly has the fine-scale Markov property since the Markov property subsumes the fine-scale Markov property. Assume, then, that $x(\cdot)$ has the fine-scale Markov property and let $s, r, t \in \mathcal{S}_0$ such that the unique shortest path from s to t goes through r. Subject only to this condition, s, r, t are arbitrary so we need to show that $x(s)$ and $x(t)$ are conditionally uncorrelated when conditioned on $x(r)$. There are two cases.

Case 1. First consider the case where t is not an ancestor of s nor is s an ancestor of t (see Figure 3.2). Formally, there does not exist an integer n such that $t = s\bar{\gamma}^n$ or $s = t\bar{\gamma}^n$. It is clear from Figure 3.2, that the index sets for the vectors x_t^M and x_s^M do not overlap. Formally, there is no set in the collection $\{\mathcal{T}_{r\alpha_i}(M)\}_{i=1}^q \cup \{\mathcal{T}_r^c(M)\}$ that contains elements of both $\mathcal{T}_t(M)$ and $\mathcal{T}_s(M)$. Therefore, by assumption, $x(r)$ conditionally decorrelates x_s^M and x_t^M. It follows that $x(r)$ conditionally decorrelates $x(s) = W_s x_s^M$ and $x(t) = W_t x_t^M$.

Case 2. Next consider the case where $s = t\bar{\gamma}^n$ for some n (see Figure 3.3). Therefore s is also an ancestor of r and consequently x_t^M is contained in x_r^M. It follows that $x(t)$ is a linear function of x_r^M so it suffices to show that $x(s)$ and x_r^M are conditionally decorrelated by $x(r)$. Because by assumption $x(\cdot)$ is an internal process, we may write $x(s)$ as

$$x(s) = W_s x_s^M = L_{sr} x_r^M + L x_{rc}^M \,, \tag{3.8}$$

for some matrices L_{sr} and L where $L_{sr} = D_{sr} W_r$, for some matrix D_{sr}. Therefore,

$$\text{rowspace}(L_{sr}) \subseteq \text{rowspace}(W_r) \,. \tag{3.9}$$

Thus, $L_{sr} x_r^M$ can be linearly estimated from $x(r) = W_r x_r^M$ without error. Also, conditioned on $x(r)$, x_{rc}^M and x_r^M are conditionally uncorrelated by hypothesis. Therefore, $x(r)$ conditionally decorrelates $x(s)$ and x_r^M. ∎

As we now develop, another notion of Markovianity which is equivalent to the Markov property is the *scale-recursive Markov property.* The development of this scale-recursive formulation of the Markov property is one of the major contributions of this thesis. It is significant because it permits us to view the stochastic realization problem scale-recursively and, thereby, develop an efficient algorithm.

Definition 3.2.2 (Scale-Recursive Markov Property). *Tree-indexed process $x(\cdot)$ has the* scale-recursive Markov property *if conditioned on $x(s)$ for any $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$ the $q+1$ vectors in the set $\{x(s\alpha_i)\}_{i=1}^q \cup \{x_{sc}^{m(s)+1}\}$ are conditionally uncorrelated.*

Referring to Figure 3.1, we see that the scale-recursive Markov property is similar to the fine-scale Markov property except that rather than focusing on the leaf-node states, it focuses on those at the preceding finer scale (in the shaded regions). Since the sets of nodes associated with the scale-recursive Markov property are asymptotically of strictly smaller order than those associated with the Markov property (solid-lined regions labeled "A", "B", and "C") or the fine-scale Markov property (dotted-lined regions), the realization algorithm based on scale-recursive Markovianity (developed in Chapter 4) is orders of magnitude more efficient than previous approaches. Specifically, at scale $M-1$, the total number of variables considered is the same for both the fine-scale Markov property and the scale-recursive Markov property. However, at coarser scales, the sets involved in the scale-recursive Markov property are smaller than those involved in the fine-scale Markov property. Indeed, at each successive coarser scale, the total number of variables considered in the scale-recursive Markov property is reduced by a factor of q. We now show that the scale-recursive Markov property is equivalent to the Markov property for internal processes.

Proposition 3.2.2. *Assume $x(\cdot)$ is an internal tree-indexed process. Then $x(\cdot)$ has the scale-recursive Markov property if and only if it has the Markov property.*

Proof. First, if $x(\cdot)$ has the Markov property then, by definition, it has the scale-recursive Markov property. Next, assume that $x(\cdot)$ has the scale-recursive Markov

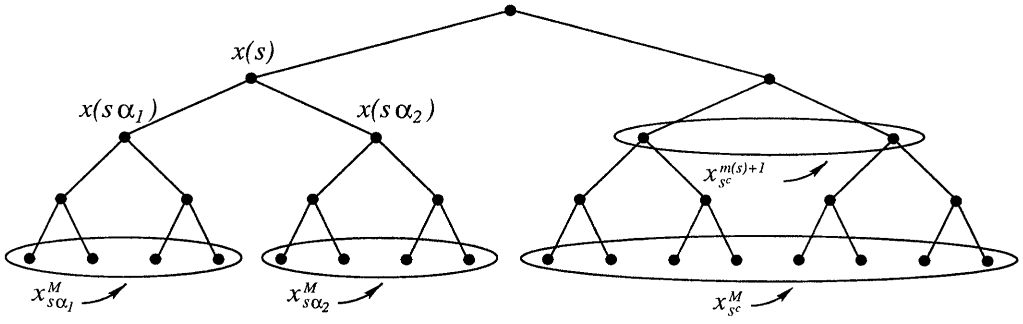


Figure 3.4. As shown in the proof of Proposition 3.2.2, if $x(s)$ conditionally decorrelates the vectors in the set $\{x(s\alpha_i)\}_{i=1}^q \cup \{x_{s^c}^{m(s)+1}\}$ then it conditionally decorrelates the vectors in the set $\{x_{s\alpha_i}^M\}_{i=1}^q \cup \{x_{s^c}^M\}$. This figure illustrates the case for $q = 2$.

property. We show in Appendix A that for an arbitrary s in $\mathcal{S}_0 - \mathcal{T}_0(M)$, $x(s)$ conditionally decorrelates the vectors in the set $\{x_{s\alpha_i}^M\}_{i=1}^q \cup \{x_{s^c}^M\}$ (see Figure 3.4). Having shown this, then $x(\cdot)$ has the fine-scale Markov property and thus, by Proposition 3.2.1, it has the Markov property. ■

■ 3.3 Implications for Stochastic Realization

The theory of internality and scale-recursive Markovianity developed in this chapter, as well as that of predictive efficiency developed in Section 2.4, forms the foundation for the remainder of this thesis. In Chapter 4 we will combine the notions of internality, scale-recursive Markovianity, and predictive efficiency in developing an efficient, general-purpose, and scale-recursive MAR model identification algorithm. In Chapter 5 we rely on the theoretical development of internality in order to design MAR states that consist of wavelet scaling and detail coefficients associated with a fine-scale process. In Chapter 6 we develop internal MAR models for the maximum-entropy extension of partially specified covariance matrices. Finally, in Chapter 7 we present several ways of incorporating nonlocal variables into internal MAR models. One of these relies on an intellectual successor to the Markov property.

We now highlight the features of the foregoing theory which are of greatest importance for the remainder of this thesis. First, previously-developed systematic MAR realization algorithms lead to model inconsistencies precisely because they are based on an incorrect parameterization of internal states, (3.1). In contrast, the models developed in this thesis are based on (3.2) which guarantees internality and consistency. In turn, internality provides the theoretical foundation for the identification of consistent MAR models with states defined by specific linear functionals like scaling and wavelet functionals or other nonlocal functionals of importance in particular estimation problems. Moreover, internality also plays an important role in overcoming the computational burden of previously-developed realization algorithms [37,40,96,98,100].

The computational complexity of these previous approaches stems from several sources, two of which have now been exposed. First, because they are based on the fine-scale Markov property and use canonical correlations, they must consider the statistics (covariance matrix) for a large number of variables, those indexed by the fine-scale nodes. Second, they must do this at *every* node. In contrast, in Chapter 4 we focus on the (equivalent) scale-recursive Markov property which alleviates the latter source of complexity as described in this chapter. We also use predictive efficiency and not canonical correlations which alleviates the former source of complexity as described in Chapter 2.

Chapter 4

Scale-Recursive Stochastic Realization

THIS chapter[1] addresses the MAR stochastic realization problem which we defined and discussed in Section 2.3. Recall that the MAR stochastic realization problem is one of choosing MAR parameters, $P_{x(0)}$, $A(\cdot)$, and $Q(\cdot)$, such that the realized fine-scale sub-process x^M has a covariance matrix P_{x^M} that well-approximates a given covariance matrix P_{f^M} associated with the fine-scale process f^M. For convenience only, it is assumed that f^M has length $N = dq^M$. In Section 4.1 we develop an algorithm to address this problem that has complexity quadratic in problem size (i.e., it is $O(N^2)$). In Section 4.2 we introduce an approximation that reduces the complexity to linear in problem size. In the examples in Section 4.3 we illustrate the performance of our methodology and the degree of error introduced by the approximation of Section 4.2.

The algorithm presented in this chapter is based on predictive efficiency, which we discussed in Section 2.4. Additionally, it relies on the theory of internality and scale-recursive Markovianity developed in Chapter 3. The development of Chapter 3 makes clear that the notion of Markovianity is related to the information content of MAR states. We have also seen that the information of internal MAR states is parameterized by local internal matrices. Hence, at the heart of our approach is the design of local internal matrices. From these, MAR model dynamics are easily derived as discussed in Section 2.3.

The algorithm developed in this chapter overcomes two weaknesses of those previously developed [37,40,96,98,100]. First, the resulting models are internal and, hence, possess states that are consistent with one another. Second, the algorithm is considerably more computationally efficient than previously-developed ones. In contrast to the quadratic (and, with an approximation, linear) complexity of our approach, the canonical correlations approach of [96,98,100] has complexity quartic in problem size while the approach of [37,40], which is applicable to self-similar processes with stationary increments, has complexity cubic in problem size.

For notational simplicity, we assume in our development that our models have constant state dimension d. However, our approach is applicable and easily generalizable

[1] Most of the material of this chapter may be found in [77].

to cases for which state dimension varies from node to node as we show by example.

4.1 Predictive Efficiency-Based Internal Realization

In this section we first present an $O(N^2)$ algorithm for internal MAR model identification (Section 4.1.1). Following that we provide some analysis (Section 4.1.2).

4.1.1 Algorithm

In this section we discuss an $O(N^2)$ algorithm for internal MAR model identification. In constructing an internal MAR model $x(\cdot)$ for P_{f^M} we define another locally internal tree-indexed (and not necessarily MAR) process as an intermediate step. This intermediate process $f(\cdot)$ has as its finest-scale sub-process f^M, the signal to be modeled. At any node s not at the finest scale, we define the value of $f(\cdot)$ at node s scale-recursively as $f(s) \triangleq V_s f_s^{m(s)+1}$ where each local internal matrix V_s is derived based on a predictive efficiency criterion (detailed shortly).

From the set of local internal matrices $\{V_s\}$ and the given fine-scale covariance P_{f^M}, the statistics $P_{f(s)}$ and $P_{f(s\alpha_i)f(s)}$ are easily computed. In turn, these may be used to define the dynamical model for $f(\cdot)$:

$$f(s\alpha_i) = A(s\alpha_i)f(s) + \mu(s\alpha_i)\,, \tag{4.1a}$$

$$Q(s\alpha_i) \triangleq \mathrm{E}[\mu(s\alpha_i)\mu(s\alpha_i)^T] \tag{4.1b}$$

where $A(s\alpha_i)$ and $Q(s\alpha_i)$ are computed from $P_{f(s)}$ and $P_{f(s\alpha_i)f(s)}$ as described in Proposition 3.1.2 and (3.7) (in which $x_s^{m(s)+1}$, $x(s\alpha_i)$, and $x(s)$ are replaced by $f_s^{m(s)+1}$, $f(s\alpha_i)$ and $f(s)$, respectively).

If the process $f(\cdot)$ has the scale-recursive Markov property then $\mu(\cdot)$ is a white noise process, uncorrelated with $f(0)$. Hence, (4.1) is an exact MAR model for f^M [100]. As we will explain, this will occur if no approximation is made in the predictive efficiency step that defines the local internal matrices. If, on the other hand, we *do* make an approximation in the predictive efficiency step, then $\mu(\cdot)$ will not be a white noise process, uncorrelated with $f(0)$. However, in this case, we can *define* an *approximate* model by assuming that $\mu(\cdot)$ is white and uncorrelated with $f(0)$. That is, we define the internal MAR process[2] $x(\cdot)$ to approximate $f(\cdot)$ as

$$x(s\alpha_i) = A(s\alpha_i)x(s) + w(s\alpha_i)\,, \tag{4.2a}$$

$$\mathrm{E}[w(s\alpha_i)w(s\alpha_i)^T] = Q(s\alpha_i) \tag{4.2b}$$

where $A(\cdot)$ and $Q(\cdot)$ are the same as in (4.1) and $w(\cdot)$ is white, uncorrelated with $x(0)$. Note that, while this results in an approximate model ($P_{x^M} \neq P_{f^M}$), the state covariances $P_{x(s)}$ at each node s and the child-parent cross-covariances $P_{x(s\alpha_i)x(s)}$ for

[2] A formal proof of internality is provided in Section 4.1.2.

each child-parent pair of nodes exactly match $P_{f(s)}$ and $P_{f(s\alpha_i)f(s)}$, respectively. Consequently, the $d \times d$ diagonal blocks of P_{x^M} exactly match those of P_{f^M}.

It remains only to specify the predictive efficiency step in which we define the local internal matrices $\{V_s\}$. To obtain an exact model requires that $f(\cdot)$ have the scale-recursive Markov property which it does (by definition) if $f(s) = V_s f_s^{m(s)+1}$ conditionally decorrelates the set of vectors $\{f(s\alpha_i)\}_{i=1}^{q} \cup \{f_{s^c}^{m(s)+1}\}$ for all $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$. As discussed in Section 2.4, this occurs exactly when

$$\bar{\varepsilon}\big(f(s\alpha_1), f(s\alpha_2), \dots, f(s\alpha_q), f_{s^c}^{m(s)+1} \mid V_s f_s^{m(s)+1}\big) = 0 . \tag{4.3}$$

Typically, any V_s satisfying (4.3) has too many rows, leading to models with impractically high state dimensions. Therefore, to obtain lower dimensional states, we may apply the procedure described in Section 2.4 to find a suboptimal solution to the predictive efficiency problem

$$\arg \min_{V \in \mathcal{M}_d^*} \bar{\varepsilon}\big(f(s\alpha_1), f(s\alpha_2), \dots, f(s\alpha_q), f_{s^c}^{m(s)+1} \mid V f_s^{m(s)+1}\big) \tag{4.4}$$

thereby constraining V_s to have no more than d rows.

The asymptotic computational complexity of our realization approach stems from two sources. The first is the computation of the local internal matrices. If d is chosen independent of problem size N then, as described in Section 2.4, the complexity of finding our suboptimal solution to (4.4) is $O(q^n)$ because $f_{s\alpha_i^c}^{m(s)+1}$ (which plays the role of z_i^c of Section 2.4), is length $O(q^n)$. Summing this up over all nodes we arrive at an $O(q^{2M})$ complexity which is equivalent to $O(N^2)$ because $N \propto q^M$. The second source of complexity is the computation of P_{f^n}, the statistics of $f(\cdot)$ at scale n which are needed to compute the local internal matrices at scale $n-1$. We have that

$$P_{f^n} = \mathcal{V}_n P_{f^{n+1}} \mathcal{V}_n^T \tag{4.5}$$

where $\mathcal{V}_n$ is a block diagonal matrix whose diagonal blocks are V_s for $s \in \mathcal{T}_0(n)$, lexicographically ordered. By construction, each row of $\mathcal{V}_n$ has at most d non-zero elements. Taking advantage of this sparsity, we can compute $\{P_{f^n}\}_{n=0}^{M-1}$ with complexity $O(N^2)$.

■ 4.1.2 Analysis

In this section we provide formal justification for the scale-recursive realization algorithm presented in the previous section. First, the fact that $f(\cdot)$ as defined by (4.1) is internal follows from the the following simple lemma.

Lemma 4.1.1. *Let Q be the covariance matrix for*

$$\mu = \big[\mu(s\alpha_1)^T \quad \mu(s\alpha_2)^T \quad \cdots \quad \mu(s\alpha_q)^T\big]^T \tag{4.6}$$

where $\mu(s\alpha_i)$ is as defined in (4.1). *Then Q has the form*

$$Q = \begin{pmatrix} Q(s\alpha_1) & Q(s\alpha_1, s\alpha_2) & \cdots & Q(s\alpha_1, s\alpha_q) \\ Q(s\alpha_2, s\alpha_1) & Q(s\alpha_2) & \cdots & Q(s\alpha_2, s\alpha_q) \\ \vdots & \vdots & \ddots & \vdots \\ Q(s\alpha_q, s\alpha_1) & Q(s\alpha_q, s\alpha_2) & \cdots & Q(s\alpha_q) \end{pmatrix} \tag{4.7}$$

and $V_s Q V_s^T = 0$.

Proof. That Q has the form of (4.7) is clear. That $V_s Q V_s^T = 0$ follows from the fact that the dynamics of $f(\cdot)$ represent the optimal prediction of $f_s^{m(s)+1}$ from $f(s) = V_s f_s^{m(s)+1}$. ■

The proof of the internality of $f(\cdot)$ follows from Lemma 4.1.1 and is nearly identical in form to that of Proposition 3.1.2.

Next, we address the internality of the MAR process $x(\cdot)$ as defined by (4.2). If the local internal matrices $\{V_s\}$ are arbitrary, then there is no guarantee that $x(\cdot)$ is internal. The problem is that the relationship $x(s) = V_s x_s^{m(s)+1}$ must be a deterministic one. However, the $x(s\alpha_i)$ have contributions from uncorrelated noises $w(s\alpha_i)$. Thus, it is necessary that the driving noise be in the null-space of V_s. As the following proposition shows, if each V_s is block diagonal (as ours are), then the relationship $x(s) = V_s x_s^{m(s)+1}$ is deterministic for all s and $x(\cdot)$ *is* internal.

Proposition 4.1.1. *Let V_s have a block structure given by*

$$V_s = \begin{pmatrix} V_{11} & V_{12} & \cdots & V_{1q} \\ V_{21} & V_{22} & \cdots & V_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ V_{q1} & V_{q2} & \cdots & V_{qq} \end{pmatrix} \tag{4.8}$$

where in the expression $V_s f_s^{m(s)+1}$, the submatrix V_{ij} acts on $f(s\alpha_j)$. If $V_{ij} = 0$ for all $i \neq j$ then $x(\cdot)$ as defined by (4.2) *is internal.*

Proof. See Appendix B. ■

We emphasize that the condition that V_s is block diagonal is a sufficient but not necessary one for $x(\cdot)$ to be internal. Indeed, by an argument similar to the one in the proof of Proposition 3.1.2, it is easily shown that it is both necessary and sufficient that $V_s R V_s^T = 0$ where $R = \text{diag}(Q(s\alpha_1), Q(s\alpha_2), \ldots, Q(s\alpha_q))$ is the covariance matrix for $\left[w(s\alpha_1)^T \;\; w(s\alpha_2)^T \;\; \cdots \;\; w(s\alpha_q)^T\right]^T$ and $w(s\alpha_i)$ is as defined in (4.2). That is, R is derived from Q (cf., (4.7)) by setting the off-diagonal blocks to zero. As we will see in Chapter 5 (in which the V_s are defined by wavelet bases), there are important cases for which $V_s R V_s^T = 0$, and yet V_s is not block diagonal. However, in general, it is unclear how to obtain $V_s R V_s^T = 0$ without restricting V_s to be block diagonal. When we *do* restrict V_s to be block diagonal, the realization procedure is particularly simple, as is evidenced by the developments of this chapter. This, then, provides substantial motivation and justification for imposing a block diagonal structure.

4.2 Boundary Approximation

The predictive efficiency-based MAR realization method proposed in the previous section has complexity proportional to N^2 (where the signal or (lexicographically ordered) image to be modeled has total size N). While this is relatively efficient as compared to other approaches [37, 40, 96, 98, 100], it is still too burdensome for some problems, particularly those arising in image processing. The source of this complexity stems from the fact that, in computing the predictive efficiency matrices, we focus on estimating *every* element of a *large* random vector, z_2, from a small one, z_1. In Section 4.2.1, we propose the *boundary approximation* which focuses on estimating only a small number of elements of z_2 which are temporally or spatially close to z_1. As we will show, this boundary approximation leads to a realization algorithm that has complexity proportional to N. We note that a similar approximation is employed in conjunction with canonical correlations in [96, 100]. However, in some sense it is more severe because, due to the symmetry of canonical correlations, it requires truncating *both* z_2 *and* z_1. Since predictive efficiency is not symmetric, we need only truncate z_2 to obtain an $O(N)$ algorithm.

Intuitively, the boundary approximation should not be a severe one for processes that are Markov (or nearly so) or have quickly decaying long-range correlations. In the former case, the boundaries of z_1 contain all the relevant information for estimating the more distant random variables. Therefore, a summary of z_1 (i.e., Vz_1) that does a good job of estimating these local variables ought to be sufficient for estimating the distant ones. In the latter case of quickly decaying long-range correlations, distant random variables are negligibly correlated with z_1 and, therefore, do not substantially contribute to the mean-square estimation error. We will make this intuition precise in Section 4.2.2. In our examples in Section 4.3, we will show that the class of processes for which the boundary approximation results in small modeling error is, in fact, considerably *richer* than Markov and fast-decorrelating processes. A theoretical understanding of this fact is a topic for future research, as we discuss in Chapter 8.

In this section we will rely on the following notation which was introduced in Section 2.4. The zero-mean vectors z_1 and z_2 have lengths n_1 and n_2 and covariances P_1 and P_2, respectively. Their cross-covariance matrix is $\mathrm{E}[z_2 z_1^T] = P_{21}$.

4.2.1 Algorithm

Let us begin by examining the two sources of the N^2 complexity of our realization approach in the context of building a MAR model for a one-dimensional random signal (as opposed to a two-dimensional random field, to which we return later). The first

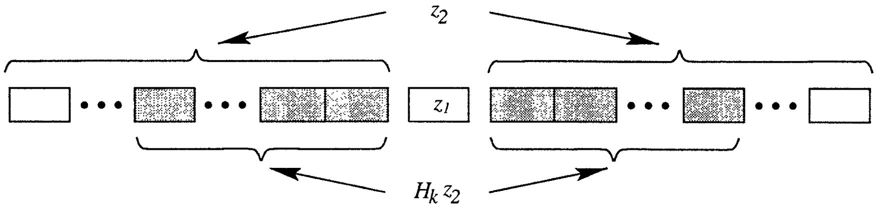


Figure 4.1. $H_k z_2$ selects the $2kd$ elements of z_2 (shaded) that are temporally closest to z_1.

source comes from finding the local internal matrices which are suboptimal solutions to (4.4). The second is the computation of P_{f^n} for all scales $n \in \{0, 1, \dots, M-1\}$. With respect to the former, we noted in Section 2.4 that summarizing a length-n_1 vector z_1 for the purposes of estimating a length-n_2 vector z_2 has complexity $O(n_2)$. In the context of the stochastic realization problem, this translated into a complexity of $O(N)$ per node which, when summed over all $O(N)$ nodes, lead to an overall $O(N^2)$ complexity for computing the internal matrices. This suggests that we can reduce the overall complexity to $O(N)$ by somehow ignoring all but a small portion of z_2 (whose size is independent of N).

To this end, let k be an integer chosen independent of n_2, and let H_k be a selection matrix such that, when post-multiplied by z_2, it selects the $2kd$ elements of z_2 that are temporally closest to z_1 as illustrated in Figure 4.1 (i.e., the kd elements on either side of z_1 are selected). We will call the $2kd$ elements selected by H_k the *size-k boundary* of z_1. With this notation, consider

$$\widehat{V}_k \triangleq \arg\min_{V \in \mathcal{M}_r} \bar{\varepsilon}(H_k z_2 \mid V z_1). \tag{4.9}$$

Since the complexity of computing $H_k P_2 H_k^T$ (a quantity needed to solve (4.9)) is independent n_2, the solution of (4.9) can be computed with complexity that is also independent of n_2. We then view $\widehat{V}_k$ as a suboptimal solution to (2.54). Using this idea in our stochastic realization approach, we arrive at a complexity of $O(N)$ for computing the internal matrices.

We now turn to the second source of the N^2 complexity of the MAR stochastic realization approach of Section 4.1—the computation of the P_{f^n}. The boundary approximation reduces this source of complexity as well since using (4.9) implies that we need not compute all of P_{f^n}. Rather, only a diagonal band of size that is a function of k is needed because we never consider cross-correlations involving elements that are further than kd away from the node at which the current predictive efficiency matrices are being computed. It is not hard to show that the total complexity of computing the required diagonal bands of the P_{f^n} matrices for all $n \in \{0, 1, \dots, M-1\}$ is $O(N)$. Hence, the overall asymptotic complexity of the MAR realization algorithm with the boundary approximation is $O(N)$.

We now discuss the boundary approximation for modeling two-dimensional random

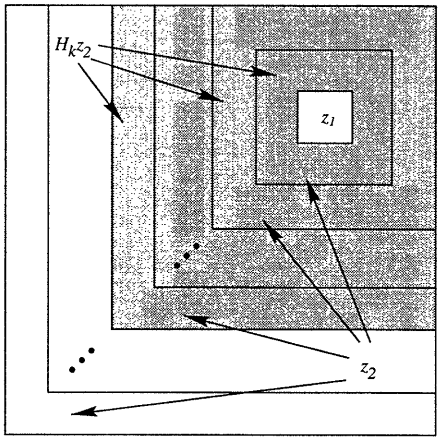


Figure 4.2. $H_k z_2$ selects the elements of z_2 (shaded) that are spatially closest to z_1, those in the k concentric square annuli around z_1.

fields using quad-trees. In this case, the vector z_1 represents a pixel[3] of a random field and z_2 the rest of the random field as illustrated in Figure 4.2. The matrix H_k selects the elements of z_2 that lie in the k concentric square annuli, each of which is one pixel wide, that surround z_1 (with the obvious modifications for boundary effects as illustrated). All of the complexity analysis provided previously for the one-dimensional case is identical for the two-dimensional case.

4.2.2 Analysis

In this section we provide some theoretical justification for the intuition that motivated the boundary approximation, namely that the boundary approximation ought not be a severe one for processes that are Markov (or nearly so) or have quickly-decaying long-range correlations. For this purpose we will rely on the notation developed in previous sections as well as the following. Let J_k be a selection matrix that complements H_k in the sense that $J_k z_2$ selects all the elements of z_2 that are outside the size-k boundary of z_1. Referring to Figure 4.1 and Figure 4.2, $J_k z_2$ selects the elements of z_2 that are not shaded. Therefore, every element of z_2 is an element of $H_k z_2$ or $J_k z_2$.

In making the boundary approximation, we choose a matrix V to minimize $\bar{\varepsilon}(H_k z_2 \mid Vz_1)$ rather than the exact criterion $\bar{\varepsilon}(z_2 \mid Vz_1)$. The two criteria are related by

$$\bar{\varepsilon}(z_2 \mid Vz_1) = \bar{\varepsilon}(H_k z_2 \mid Vz_1) + \bar{\varepsilon}(J_k z_2 \mid Vz_1). \tag{4.10}$$

Therefore, for a given V, one measure of the quality of the boundary approximation is the value of $\bar{\varepsilon}(J_k z_2 \mid Vz_1)$. Since $\bar{\varepsilon}(\cdot \mid \cdot\cdot)$ is lower-bounded by zero, the boundary approximation is, in fact, equivalent to the exact implementation (cf., (2.54)) when

[3] In a quad-tree, each scale is composed of pixels. At coarser scales the state vector at each pixel (here written as z_1) provides an abstract summary of a multi-pixel region of the fine-scale image.

$\bar{\varepsilon}(J_k z_2 \mid \widehat{V}_k z_1) = 0$. In the two propositions of this section, we bound the quantity $\bar{\varepsilon}(J_k z_2 \mid V z_1)$ given certain conditions. We first consider processes whose long-range correlations are small. That is, the correlations between elements of z_1 and elements of $J_k z_2$ are, in some sense, small.

Proposition 4.2.1. *Let λ be the maximum singular value of* $\operatorname{cov}(J_k z_2, z_1) = J_k P_{21}$ *and let*

$$\alpha \triangleq \max_j \left\| P_1^{-T/2}(:, j) \right\| . \tag{4.11}$$

Then $\bar{\varepsilon}(J_k z_2 \mid V z_1) \leq n_1 \alpha^2 \lambda^2$.

Proof. See Appendix B. ∎

There are a number of important points to emphasize regarding Proposition 4.2.1.

- As the long-range correlations (those between z_1 and $J_k z_2$) go to zero, λ goes to zero and $\bar{\varepsilon}(J_k z_2 \mid V z_1)$ goes to zero.
- The bound holds for all matrices V and, therefore, holds for the optimal choice under the boundary approximation $\widehat{V}_k$ as defined by (4.9).
- The bound is loose and depends on n_1. No doubt tighter bounds can be obtained.

Proposition 4.2.1 is intended to provide some theoretical justification for the boundary approximation when the covariance matrix for the underlying process is exactly or approximately banded. When the *inverse* of the covariance matrix is exactly or approximately banded, we are dealing with a process that is exactly or approximately Markov. For nearly k-th order Markov processes, $H_k z_2$ approximately conditionally decorrelates z_1 and $J_k z_2$. Therefore, the entries of

$$\Delta \triangleq \mathrm{E}\left[J_k z_2 (z_1 - \widehat{\mathrm{E}}[z_1 \mid H_k z_2])^T\right] = J_k P_{21} - J_k P_2 H_k^T (H_k P_2 H_k^T)^{-1} H_k P_{21} \tag{4.12}$$

will be small. The following proposition shows that, in this case, the quantity $\bar{\varepsilon}(J_k z_2 \mid V z_1)$ is related to two sources of error. One source stems from the degree to which the process can be approximated by a k-th order Markov process and is related to the size of the entries of Δ. The other source stems from how well we are able to estimate the boundary $H_k z_2$ from the summary of z_1 given by $V z_1$. That is, the second source of error is related to $\bar{\varepsilon}(H_k z_2 \mid V z_1)$.

Proposition 4.2.2. *Let δ be the maximum singular value of Δ as defined in* (4.12) *and let $\bar{\varepsilon}(H_k z_2 \mid V z_1) < \sigma^2$. Let λ be the maximum singular value of*

$$\Lambda \triangleq \left(P_1^{-1} - V^T (V P_1 V^T)^{-1} V\right) P_{21}^T H_k^T (H_k P_2 H_k^T)^{-1} H_k P_2 J_k^T . \tag{4.13}$$

Then

$$\bar{\varepsilon}(J_k z_2 \mid V z_1) \leq n_1(\alpha^2 \delta^2 + 2|\delta\lambda|) + n_3 \beta^2 \sigma^2 \tag{4.14}$$

where

$$\alpha \triangleq \max_j \left\| \left[(P_1^{-1} - V^T(VP_1V^T)^{-1}V)^{1/2} \right]_{:,j} \right\| , \tag{4.15a}$$

$$\beta \triangleq \max_j \left\| \left[(H_kP_2H_k^T)^{-1}H_kP_2J_k^T \right]_{:,j} \right\| . \tag{4.15b}$$

and n_3 is the length of the vector J_kz_2.

Proof. See Appendix B. ■

It is worth noting the following regarding Proposition 4.2.2.

- The interpretations of the matrix Λ and the variables α and β are unclear. A more precise theoretical justification for the boundary approximation is an open problem.

- The first term of the upper bound, $n_1(\alpha^2\delta^2 + 2|\delta\lambda|)$, is related to the degree to which the process is k-th order Markov. It depends on δ which is small when the process is nearly k-th order Markov and is zero if the process is exactly k-th order Markov.

- The second term of the upper bound, $n_3\beta^2\sigma^2$, is related to how well the information contained in Vz_1 performs in estimating the boundary H_kz_2. It depends on σ^2 which bounds $\bar{\varepsilon}(H_kz_2 \mid Vz_1)$. Notice that this term is indirectly related to the number of linear functionals used for summarizing z_1 (rows of V) since, as the number of (linearly independent) linear functionals increases, $\bar{\varepsilon}(H_kz_2 \mid Vz_1)$ decreases leading to a smaller bound σ^2.

- The bound holds for all matrices V and, therefore, holds for the optimal choice under the boundary approximation $\widehat{V}_k$ as defined by (4.9).

- The bound is loose and depends on n_1 and n_3. Tighter bounds can likely be obtained.

- Note that, for one-dimensional processes, $n_3 = n_2 - 2dk$ so that the n_3 can be eliminated and the bound may be expressed in terms of n_1, n_2, d, and k as

$$\bar{\varepsilon}(J_kz_2 \mid Vz_1) \leq n_1(\alpha^2\delta^2 + 2|\delta\lambda|) + (n_2 - 2dk)\beta^2\sigma^2 . \tag{4.16}$$

For two-dimensional processes, a similar expression can be found where n_2 is n_3 less the number of elements in k concentric, one-pixel wide annuli (ignoring boundary effects).

In this section we have provided some formal justification for the intuition behind the boundary approximation. Nevertheless, Proposition 4.2.1 and Proposition 4.2.2 have a number of limitations. In addition to those stated, the explanatory power of

Proposition 4.2.1 and Proposition 4.2.2 is limited for two reasons. First, these two propositions rely on the statistical structure of the underlying signal $z \triangleq \begin{bmatrix} z_1^T & z_2^T \end{bmatrix}^T$. While it is, perhaps, easily verified that the fine-scale process to be modeled, f^M, has the required structure (quickly decaying long-range correlations or approximate Markovianity), whatever statistical structure is present in f^M is typically not preserved at coarser scales. That is, the statistical properties of coarser-scale variables are derived from f^M *filtered* by the linear functionals that define the internal matrices. Therefore, if f^M is Markov, say, it is by no means clear that f^n for $n < M$ is Markov or even approximately so.

The second limitation of Proposition 4.2.1 and Proposition 4.2.2 is that they focus on the error caused by a single application of the boundary approximation. However, in in building an entire MAR model, many applications are made (specifically, q at each node). It is not clear how the bounds of Proposition 4.2.1 and Proposition 4.2.2 (which pertain to local errors) translate into the overall quality of a model. We will return to these limitations in Chapter 8.

■ 4.3 Examples

In this section we provide several examples illustrating the performance of the $O(N^2)$ realization algorithm of Section 4.1 as well as the boundary approximation discussed in Section 4.2.

■ 4.3.1 One-Dimensional Processes

Fractional Brownian Motion

Our first example is the realization and estimation of fractional Brownian motion (discussed in Section 2.3.1) with Hurst parameter $H = 0.7$ (denoted fBm(0.7)). The true fBm(0.7) covariance matrix, P_{f^M}, associated with 128 samples of fBm(0.7) on $(0, 1]$ is illustrated in Figure 4.3(a). The realized covariance matrix, P_{x^M}, associated with a MAR model with state dimension $d = 4$ and based on our full $O(N^2)$ algorithm is illustrated in Figure 4.3(b). In Figure 4.3(c) we have plotted $|P_{f^M} - P_{x^M}|$ where $|\cdot|$ is element-wise. Notice that even for this relatively low dimensional model, the approximation is quite good, with the largest element-wise error on the order of 10^{-3}. In addition, the fact that the 4×4 diagonal blocks of $|P_{f^M} - P_{x^M}|$ are zero can be plainly seen in Figure 4.3(c). Notice also that some of the largest errors correspond to correlations between elements that are spatially close. This is due to the fact that spatially close elements (like those at sample numbers 64 and 65) can be quite far apart in tree distance and the correlation between them suffers from errors induced by the approximation made at all the tree nodes between them.

In Figure 4.3(d), we have plotted the realized covariance, P_{x^M}, based on a MAR model for fBm(0.7), again with state dimension $d = 4$, but derived using the boundary approximation. The boundary size is $k = 1$ which corresponds to designing local internal matrices to (approximately) conditionally decorrelate variables at a given node from

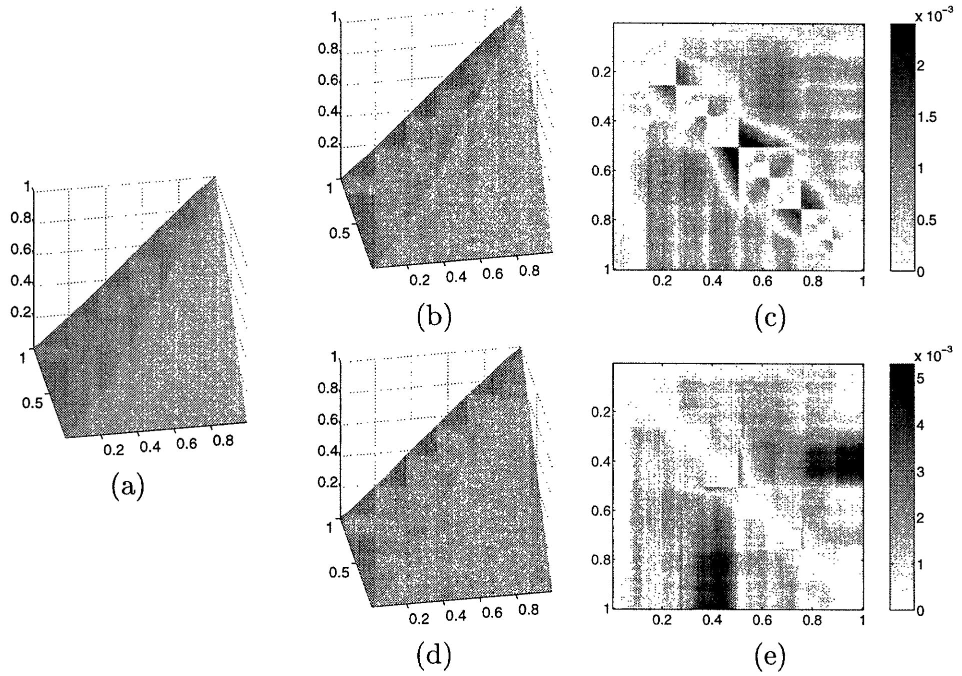


Figure 4.3. Realization of 128 samples of fBm(0.7) on $(0, 1]$. (a) Exact covariance, P_{fM}. (b) Realized covariance, P_{xM} ($d = 4$). (c) $|P_{fM} - P_{xM}|$ where P_{xM} is from (b). (d) Realized covariance, P_{xM}, using the boundary approximation ($d = 4$, $k = 1$). (e) $|P_{fM} - P_{xM}|$ where P_{xM} is from (d).

those indexed by the two nearest nodes at the same scale (or one nearest node if the given node is on the boundary). The modeling error $|P_{fM} - P_{xM}|$ is illustrated in Figure 4.3(e) and should be compared with Figure 4.3(c). Notice that the errors, while different, are of the same order, 10^{-3}. Since fBm(0.7) is not Markov and has slowly (polynomially) decaying correlations [17], this illustrates that the boundary approximation is effective for a *broader* class of processes than those that motivated it.

Next, we apply the MAR model for fBm(0.7) associated with Figure 4.3(b) to an estimation problem based on incomplete measurements corrupted by nonstationary noise. We emphasize that this is a problem that *cannot* be handled with fast transform techniques due to the nonstationarity of the process to be estimated and the process noise and the fact that the measurements are incomplete. Figure 4.4(a) is a sample-path of fBm(0.7) based on the exact statistics.[4] Figure 4.4(b) illustrates noisy, incomplete measurements of Figure 4.4(a). Measurements are taken over the first and last third of the interval $(0, 1]$. No measurements are available over the middle third. The white measurement noise has variance 0.3 over the first third sub-interval and 0.5 over the last third sub-interval. Figure 4.4(c) shows the output of the MAR estimator [29] based on

[4] Exact realizations of fBm are obtained by multiplying white Gaussian noise by the matrix square root of P_{fM}. This requires $O(N^2)$ computations if P_{fM} is $N \times N$. In contrast, the MAR sample-path generator is $O(N)$.

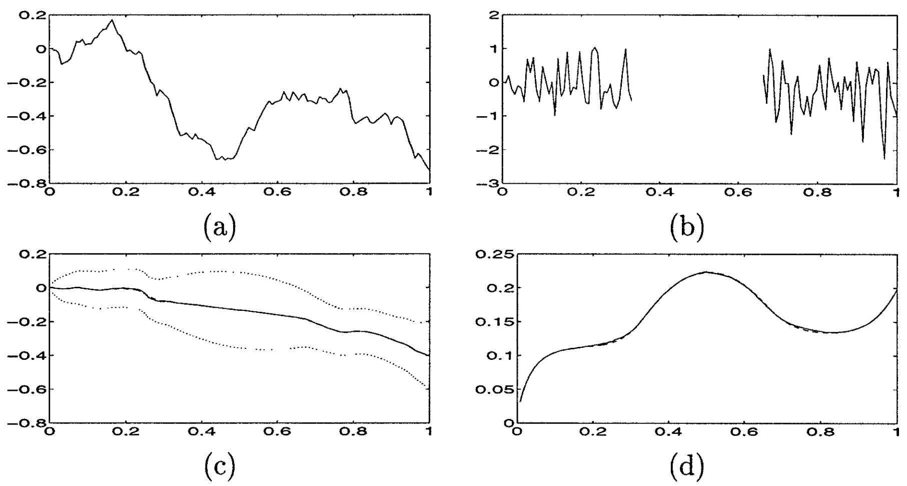


Figure 4.4. Estimation of fBm(0.7) using the model of Figure 4.3(b). (a) Sample-path using exact statistics. (b) Noisy, incomplete observations of (a). (c) MAR estimates (solid line), optimal estimates based on the exact statistics (dashed line), and plus/minus one standard deviation error bars (dotted lines). (d) Error standard deviation given by the MAR estimator (solid line) and based on the exact statistics (dashed line).

the model associated with Figure 4.3(b) (solid line) with one standard deviation error bars (dotted lines). The optimal estimate based on the exact fBm(0.7) statistics (rather than our approximate model of them) is also plotted (dashed line) in Figure 4.4(c).[5] However it is not easily distinguishable from the MAR estimate since the two nearly coincide. Moreover, the difference between the two is well within the one standard deviation error bars. This demonstrates that the degree to which our MAR model deviates from the exact model is statistically irrelevant. The MAR estimator also produces estimation error statistics with no additional computations beyond what are needed to compute the estimates themselves. In Figure 4.4(d) we have plotted the MAR error standard deviations (solid line) and the optimal error standard deviations (dashed line). The two nearly coincide, again illustrating that the degree to which our model deviates from an exact one is not relevant to this estimation problem.

Markov Process

In our next example, we illustrate MAR realizations using our $O(N^2)$ algorithm for a 12-th order stationary Markov process. The purpose of this example is to show that, while fBm can be well modeled with state dimension $d = 4$, some processes require a higher state dimension. In Figure 4.5(a) we illustrate the true covariance matrix, P_{fM}. Figure 4.5(b) illustrates the realized covariance matrix, $P_{\mathcal{X}M}$, associated with a

[5] The optimal estimates are obtained by solving the normal equations based on the true fBm and measurement statistics. Note that solving the normal equations requires $O(N^3)$ computations while the MAR estimator is $O(N)$ where N is the size of the signal to be estimated [28–30].

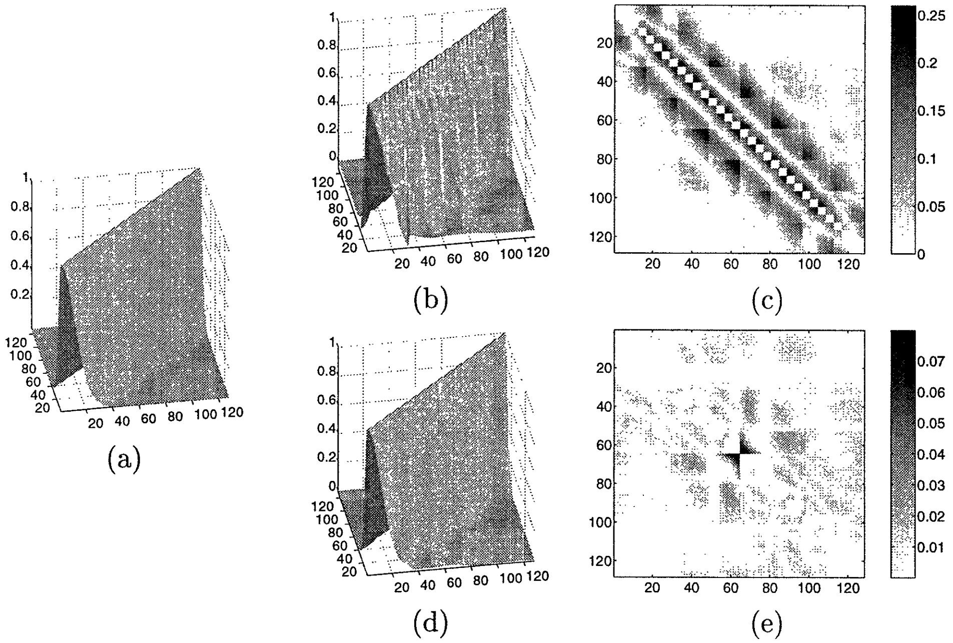
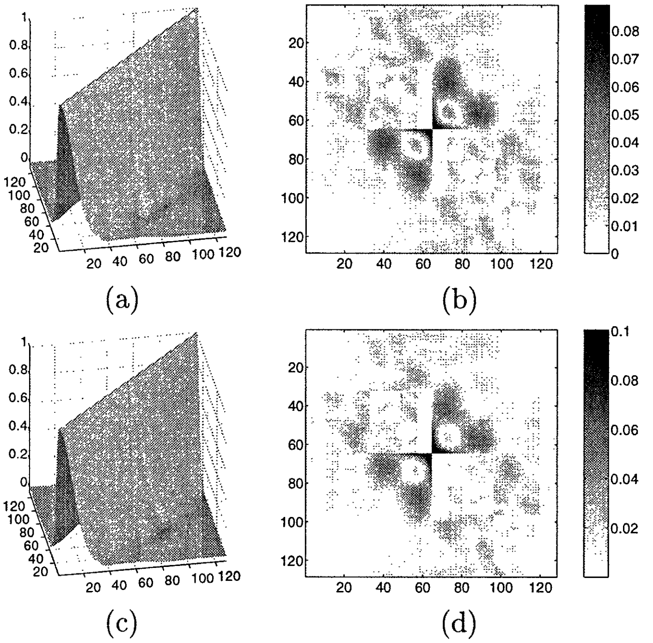


Figure 4.5. Realization of 128 samples of a 12-th order stationary Markov process. (a) Exact covariance, P_{fM}. (b) Realized covariance, P_{xM} ($d = 4$). (c) $|P_{fM} - P_{xM}|$ where P_{xM} is from (b). (d) Realized covariance, P_{xM}, ($d = 8$). (e) $|P_{fM} - P_{xM}|$ where P_{xM} is from (d).

MAR model with state dimension $d = 4$. Notice that the errors $|P_{fM} - P_{xM}|$, which are plotted in Figure 4.5(c), are *much* larger (25% of the process variance) than those associated with the fBm(0.7) model of Figure 4.3(c) which also has state dimension $d = 4$. If, however, we increase the state dimension to $d = 8$, we achieve a MAR realization with errors on the order of 7% of the process variance. This is illustrated in Figure 4.5(d) which shows P_{xM} and Figure 4.5(e) which shows $|P_{fM} - P_{xM}|$ for this higher state dimension model. A more accurate model of the 12-th order stationary Markov process than the one associated with Figure 4.5(d) requires a maximum state dimension larger than $d = 8$.

To achieve modeling errors on the order of those depicted in Figure 4.5(e), one need not use a model with state dimension $d = 8$ at all nodes. It is possible to achieve similar performance with state dimensions that decrease at coarser scales. We illustrate this point in Figure 4.6. Figure 4.6(a) is the realized covariance matrix, P_{xM}, associated with a four-scale MAR model with state dimension 8 at scales 3 (the finest) and 2, state dimension 6 at scale 1, and state dimension 4 at scale 0 (the coarsest). The error $|P_{fM} - P_{xM}|$ is plotted in Figure 4.6(b) and is on the order of 8% of the process variance, comparable to that achieved with the $d = 8$ (at all nodes) model of Figure 4.5(e).

Figure 4.6(c) illustrates the realized covariance for another MAR model of the 12-th order stationary Markov process with state dimensions that vary with scale as described. However, in this case, the boundary approximation was used with boundary size $k =$



(a) (b) (c) (d)

Figure 4.6. Realization of 128 samples of a 12-th order stationary Markov process. (a) Realized covariance, P_{x^M} (state dimension varies with scale (see text)). (b) $|P_{f^M} - P_{x^M}|$ where P_{x^M} is from (a). (c) Realized covariance using boundary approximation ($k = 3$). (d) $|P_{f^M} - P_{x^M}|$ where P_{x^M} is from (c).

3. Errors are plotted in Figure 4.6(d) and should be compared with Figure 4.6(b). Notice that the errors, while slightly different, are on the same order (roughly 10% of the process variance). This illustrates that little modeling fidelity is lost in making the boundary approximation. In this case, this result is consistent with our intuition because the underlying process is 12-th order Markov and a boundary size $k = 3$ corresponds to keeping kd state elements on either side of the node being designed. In this example d varies from 4 to 8 so the number of boundary elements is always at least as large as the Markov order. However, this intuition is deceptive because, as discussed at the end of Section 4.2.2, the fact that f^M is Markov does not guarantee that f^n is Markov for $n < M$. The results depicted in Figure 4.6(b) and Figure 4.6(d) are different because we are designing internal matrices to do different jobs. In the former case, we are attempting to conditionally decorrelate MAR variables at a given node from *all* other variables at the same scale. In the latter case, we are only considering the nearby variables at the same scale. Naturally, these two criteria lead to a different emphasis and different linear functionals that comprise the internal matrices.

Next we illustrate model fidelity as a function of boundary size. We again consider MAR models for the 12-th order stationary Markov process where the state dimension varies with scale as described previously. For different boundary sizes $k \in \{1, 2, 3, 4, 5, 6\}$ we computed a realization. We then compared the realized covariance to the true one

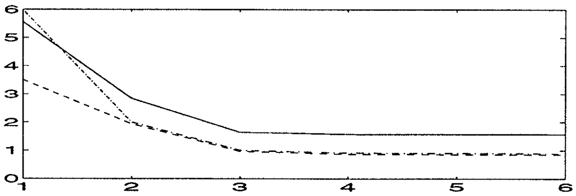


Figure 4.7. Boundary approximations for 12-th order stationary Markov process. $||P_{fM} - P_{xM}||$ is plotted as a function of boundary size k for three different norms: Frobenius (solid line), maximum singular value (dashed line), maximum element-wise absolute difference (dash-dot line). The last of these is multiplied by 10 so that it is on the same scale as the first two.

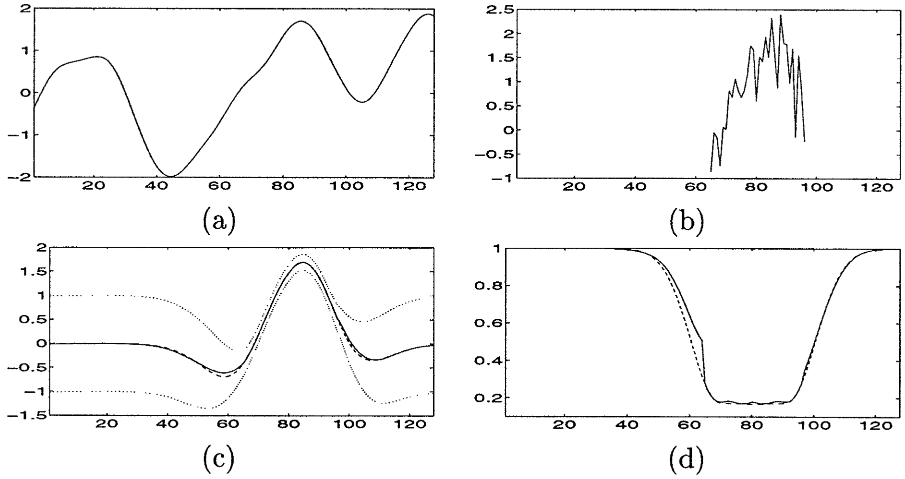


Figure 4.8. Estimation of a 12-th order stationary Markov process using the model of Figure 4.6(a). (a) Sample-path using exact statistics. (b) Noisy, observations of (a) over [65 : 96]. (c) MAR estimates (solid line), optimal estimates based on the exact statistics (dashed line), and plus/minus one standard deviation error bars (dotted lines). (d) Error standard deviation given by the MAR estimator (solid line) and based on the exact statistics (dashed line).

with three different norms $||P_{fM} - P_{xM}||$: the Frobenius norm, induced 2-norm (maximum singular value), and maximum absolute value of the difference $|P_{fM} - P_{xM}|$. We point out that, in our realization procedure, we are not explicitly minimizing any of these norms. Figure 4.7 illustrates the value of these three norms as a function of boundary size. As expected, modeling fidelity improves as boundary size increases. Notice that boundary size $k = 3$ seems to be the appropriate choice under these norms since negligible improvement can be expected for larger sizes and substantial degradation obtains for smaller sizes.

As pointed out previously, the most significant modeling errors occur for samples that are close spatially but distant on the tree. In our next example, we explore the impact of this phenomena on an estimation problem that is, in some sense, most likely to test this modeling weakness. Figure 4.8(a) is a sample-path of a 12-th order stationary Markov process. Figure 4.8(b) illustrates noisy and incomplete measurements of Figure 4.8(a). Measurements are taken only over the interval [65 : 96] which is just to the right of the largest tree boundary and the point of greatest modeling error. The white measurement noise has variance 0.3. Figure 4.8(c) shows the output of the MAR estimator based on the model associated with Figure 4.6(a) (solid line) with one standard deviation error bars (dotted lines). The optimal estimate based on the exact statistics is also plotted (dashed line) in Figure 4.8(c). We can see that the largest estimation error due to modeling occurs just to the left of the largest tree boundary (left of sample 64) as expected given the pattern of modeling error in Figure 4.6(b) and our measurement locations. Nevertheless, the differences between the optimal and the MAR estimates are well within the one standard deviation error bars and, therefore, are not particularly significant statistically. In Figure 4.8(d) we have plotted the MAR error standard deviations (solid line) and the optimal error standard deviations (dashed line). Again, the most significant errors are just to the left of sample 64 as expected and are small.

■ 4.3.2 Two-Dimensional Processes

Wood Texture

We now turn to some image processing examples. First, we consider building a MAR model for a Markov random field that mimics the texture of wood [112]. An exact 64×64 sample-path[6] is illustrated in Figure 4.9(a). Notice that this wood texture is highly correlated vertically and less so horizontally. Figure 4.9(b) is a sample-path generated by a MAR model with state dimension $d = 16$. A distracting blockiness is apparent in this figure and is due to the quad-tree structure of our model and the small state dimension. Additionally, the extreme directionality of the wood texture makes this blockiness particularly easy to see. In some applications such blockiness is of no practical significance while in others, such as surface reconstruction where gradients must be taken [69], smoothness is required.

There are two techniques for reducing blockiness. One is to increase the state dimension. This is illustrated in Figure 4.9(c) which is a sample-path based on a MAR model with state dimension $d = 64$. Unfortunately, increasing the state dimension leads to less efficient image processing algorithms. However, there is another approach: the overlapping tree approach [63, 96, 97] discussed in Section 2.3.4. A sample-path for a MAR model based on this overlapping approach with state dimension $d = 16$ is illustrated in Figure 4.9(d). Finally, Figure 4.9(e) represents a sample image from a model constructed again using the overlapping approach but in this case also employing the boundary approximation. The boundary size is $k = 1$ which corresponds to conditionally decorrelating MAR variables with those residing at nodes one pixel away. Notice that there are no blocky artifacts in either Figure 4.9(d) or Figure 4.9(e), and both models produce wood textures comparable to that in Figure 4.9(a).

[6] To compute exact sample-paths for random fields we use the FFT techniques described in [56]. Note that this requires $O(N \log N)$ computations while MAR sample-path generation is $O(N)$.

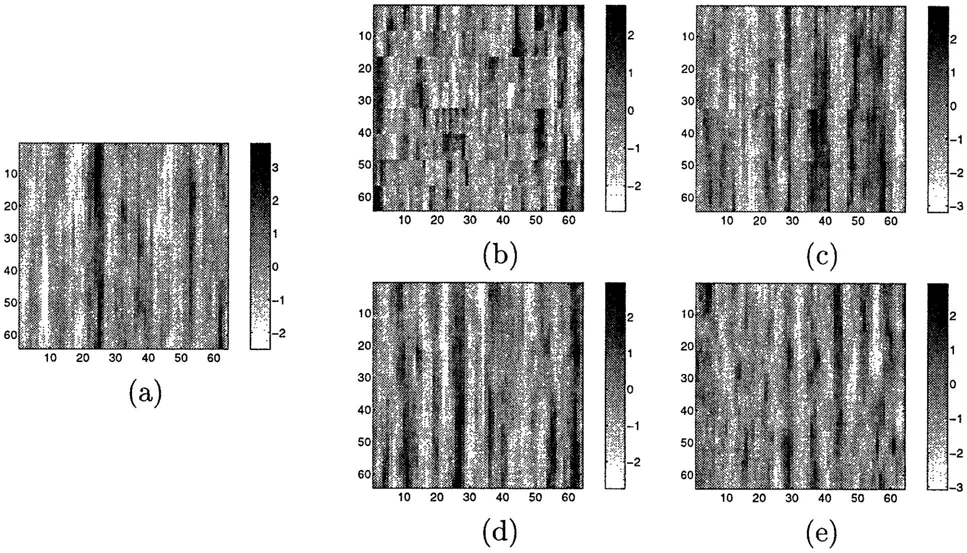


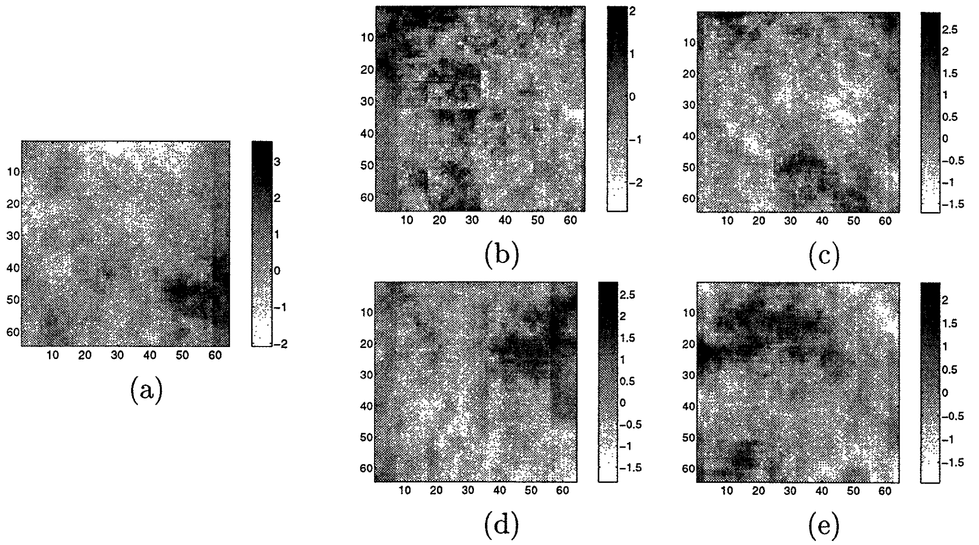
Figure 4.9. Sample-paths for wood texture of [112]. (a) Exact. (b) MAR model ($d = 16$). (c) MAR model ($d = 64$). (d) MAR model based on the overlapping tree approach ($d = 16$). (e) MAR model based on the overlapping tree approach with boundary approximation ($d = 16$, $k = 1$).

Isotropic Random Field

Next we consider sample-path generation of a two-dimensional, isotropic random field of interest in the geological sciences [103, 169]. The correlation function is

$$r_\ell(\varrho) = \begin{cases} 1 - \frac{3\varrho}{2\ell} + \frac{\varrho^3}{2\ell^3} & \text{if } 0 \leq \varrho \leq \ell, \\ 0 & \text{if } \varrho > \ell \end{cases} \tag{4.17}$$

where $\varrho = \sqrt{i^2 + j^2}$ and i, j are indices into a two-dimensional grid. An exact sample-path for $\ell = 40$ is illustrated in Figure 4.10(a). In Figure 4.10(b) and Figure 4.10(c) we provide a sample-path associated with a MAR model with state dimension $d = 16$ and $d = 64$, respectively. In Figure 4.10(d) the overlapping tree approach is used with $d = 16$. Finally, in Figure 4.10(e), the boundary approximation is employed with boundary size $k = 1$ in conjunction with the overlap method ($d = 16$). As in the previous example, little degradation is evident when the boundary approximation is used.



(a) (b) (c) (d) (e)

Figure 4.10. Sample-paths for the isotropic random field of (4.17). (a) Exact. (b) MAR model ($d = 16$). (c) MAR model ($d = 64$). (d) MAR model based on the overlapping approach ($d = 16$). (e) MAR model based on the overlapping approach with boundary approximation ($d = 16$, $k = 1$).

Chapter 5

MAR-Wavelet Processes

THIS chapter provides a unification of the MAR framework with wavelets.[1] Although the structure and development of MAR processes was motivated by and modeled on wavelet synthesis, until now the two frameworks had only been unified in the simplest case of the Haar wavelet. Recall from Section 2.3.1 that in the case of the Haar wavelet this unification is quite simple since wavelet and scaling functions do not overlap. In contrast, incorporating compactly supported and overlapping orthogonal or biorthogonal wavelets into the MAR framework is less straightforward—particularly so if we insist on internality. After a review of wavelets in Section 5.1, we show, in Section 5.2, how to build an internal MAR-wavelet process based on any compactly supported wavelet.

After developing MAR-wavelet processes, we apply them to the problem of model identification in Section 5.3. In doing so, we provide a new view of the stochastic realization problem. Previous approaches to the stochastic realization problem (including those of the preceding chapters of this thesis) focussed on designing internal matrices that define MAR states to, in some sense, optimally match the statistics of the finest scale process being modeled. As a consequence, the resulting states typically have no discernible structure beyond the fact that they represent solutions to specific optimization problems. The approach of this chapter differs in that the design of internal matrices is not closely tied to the intricate details of the fine-scale statistics. The philosophy which, in part, motivated this work is to restrict the class of linear functionals that define internal matrices to the small but rich class of wavelet bases. We thus force the states to contain meaningful multiscale representations of the fine-scale process and avoid the computationally burdensome search over all possible linear functionals. On the other hand, the approach in this chapter does bear some resemblance to previous work on the MAR stochastic realization problem: dynamics of internal MAR-wavelet models are derived from the fine-scale statistics by exploiting the linear relationship between coarse-scale and fine-scale states given by the internal matrices.

In order to be consistent with the conventional wavelet notation and to simplify our presentation, in this chapter we adopt a slightly different notation for referring to tree-indexed variables. This notation is summarized in Table 5.1 in which the scale of

[1] Most of the material in this chapter can be found in [43, 44].

Variable	*Other Chapters*	*This Chapter*
tree node	$s = (m(s), \imath(s))$	(j, n)
parent node	$s\bar{\gamma} = \big(m(s) - 1, \lfloor \imath(s)/2 \rfloor\big)$	$\big(j - 1, \lfloor n/2 \rfloor\big)$
left child	$s\alpha_1 = \big(m(s) + 1, 2\imath(s)\big)$	$\big(j + 1, 2n\big)$
right child	$s\alpha_2 = \big(m(s) + 1, 2\imath(s) + 1\big)$	$\big(j + 1, 2n + 1\big)$
state vector	$x(s)$	$x_j(n)$
process noise	$w(s)$	$w_j(n)$
autoregression matrix	$A(s)$	$A_j(n)$
process noise covariance	$Q(s)$	$Q_j(n)$
internal matrix	W_s	$W_j(n)$
local internal matrix	V_s	$V_j(n)$

Table 5.1. Notational conventions for Chapter 5.

node $s = (m(s), \imath(s))$ is denoted by $j \triangleq m(s)$ and the shift of node s is denoted by $n \triangleq \imath(s)$.

■ 5.1 Wavelet Background

This section provides a brief review of wavelets; additional details may be found in [31, 32, 46–48, 133, 134, 141, 156, 173–175]. The wavelet representation of a continuous-time signal $\vartheta(t)$ consists of a sequence of approximations of $\vartheta(t)$ at coarser and coarser scales where the approximation at the j-th scale consists of a weighted sum of shifted and dilated versions of a basic function ϕ called the *scaling function.* By considering the incremental details added in obtaining the j-th scale approximation from the approximation at scale $j - 1$, one arrives at the wavelet transform based on a single function ψ called the *analyzing wavelet.*

The reconstruction is performed using the function $\widetilde{\psi}$, called the *synthesis wavelet*, such that the two families $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ and $\{\widetilde{\psi}_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ are a biorthogonal Riesz basis of $L^2(\mathbb{R})$ where $\psi_{j,n}(t) \triangleq \sqrt{2^j}\psi(2^j t - n)$ and similarly for $\widetilde{\psi}_{j,n}$. The synthesis wavelet $\widetilde{\psi}$ is obtained from the function $\widetilde{\phi}$ which is dual to ϕ, i.e., which satisfies

$$\big\langle \phi(t), \widetilde{\phi}(t - n) \big\rangle = \delta(n) \tag{5.1}$$

where $< \cdot >$ is the standard inner product in $L^2(\mathbb{R})$ and $\delta(\cdot)$ is the discrete-time Dirac function given by

$$\delta(n) = \begin{cases} 1 & \text{if } n = 0\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.2}$$

The scaling functions ϕ and $\widetilde{\phi}$ must satisfy

$$\phi(t) = \sqrt{2}\sum_n h(n)\phi(2t-n)\,, \tag{5.3a}$$

$$\widetilde{\phi}(t) = \sqrt{2}\sum_n \widetilde{h}(n)\widetilde{\phi}(2t-n) \tag{5.3b}$$

where h and $\widetilde{h}$ are discrete filters satisfying the biorthogonality condition in $\ell^2(\mathbb{Z})$

$$\sum_k h(k)\widetilde{h}(k-2n) = \sum_k \widetilde{h}(k)h(k-2n) = \delta(n)\,. \tag{5.4}$$

The wavelets ψ and $\widetilde{\psi}$ are given by

$$\psi(t) = \sqrt{2}\sum_n g(n)\phi(2t-n)\,, \tag{5.5a}$$

$$\widetilde{\psi}(t) = \sqrt{2}\sum_n \widetilde{g}(n)\widetilde{\phi}(2t-n) \tag{5.5b}$$

where

$$g(n) = (-1)^{1-n}\widetilde{h}(1-n)\,, \tag{5.6a}$$

$$\widetilde{g}(n) = (-1)^{1-n}h(1-n)\,. \tag{5.6b}$$

The discrete filters h, g, $\widetilde{h}$ and $\widetilde{g}$ must satisfy the perfect reconstruction condition which can be found in [134]. When $h = \widetilde{h}$ and $g = \widetilde{g}$, then h is a conjugate mirror filter and the family $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ constitutes an orthonormal wavelet basis of $L^2(\mathbb{R})$.

The fast wavelet transform computes the wavelet coefficients of a discrete signal. The fast wavelet decomposition algorithm is

$$a_j(n) = \sum_p h(p-2n)a_{j+1}(p)\,, \tag{5.7a}$$

$$d_j(n) = \sum_p g(p-2n)a_{j+1}(p)\,. \tag{5.7b}$$

The reconstruction algorithm is

$$a_{j+1}(n) = \sum_p \widetilde{h}(n-2p)a_j(p) + \sum_p \widetilde{g}(n-2p)d_j(p)\,. \tag{5.8}$$

The variables $a_j(n)$ and $d_j(n)$ are called, respectively, the *scaling* and *detail coefficients* at the j-th scale and n-th shift. In Section 2.3.1 we discussed (5.7) and (5.8) for the special case of the Haar wavelet, the latter of which is illustrated in Figure 5.1.

In the remainder of this chapter, we consider only the case when h and $\widetilde{h}$ are finite impulse response (FIR) filters, i.e., when they have a finite number of non-zero

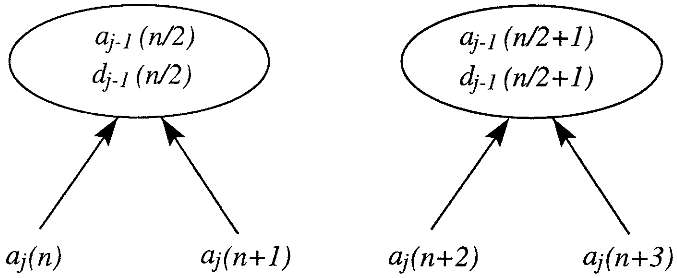


Figure 5.1. The Haar dependency graph is a dyadic tree. Here, n is even.

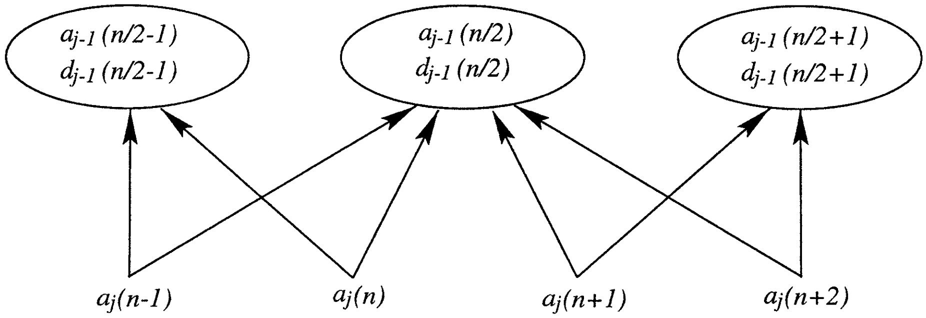


Figure 5.2. Dependency graph for the Daubechies 4-tap filter. Here n is even.

coefficients. For sake of notational simplicity, we assume that the lengths of h and $\widetilde{h}$ are both even. Without loss of generality, we choose

$$\operatorname{supp}(h) = [-R+1:R] \quad \text{and} \quad \operatorname{supp}(\widetilde{h}) = [-\widetilde{R}+1:\widetilde{R}] \tag{5.9}$$

for some integers R and $\widetilde{R}$ such that $\widetilde{R} \geq R \geq 1$. Thus, using (5.6) we have

$$\operatorname{supp}(\widetilde{g}) = [-R+1:R] \quad \text{and} \quad \operatorname{supp}(g) = [-\widetilde{R}+1:\widetilde{R}]\,. \tag{5.10}$$

We also assume that R and $\widetilde{R}$ have the same parity. We point out, however, that all the results in this chapter hold for all perfect reconstruction FIR filters with minor modifications.

The wavelet reconstruction algorithm (5.8) defines a dynamical relationship between the scaling coefficients $a_j(n)$ at one scale and those at the next finer scale, with the detail coefficients $d_j(n)$ acting as the input. Note that these dynamics are with respect to *scale* rather than time. This suggests that it is natural to think of constructing MAR processes within the wavelet framework. This construction is, in fact, obvious in the case of the Haar wavelet because each scaling coefficient depends only on time-synchronous parents[2] as discussed in Section 2.3.1 and illustrated in Figure 5.1. The link between MAR processes and wavelets is not obvious if one considers wavelets other than those

[2] The time-synchronous parents of $a_j(n)$ (or $d_j(n)$) are $a_{j-1}(\lfloor n/2 \rfloor)$ and $d_{j-1}(\lfloor n/2 \rfloor)$.

in the Haar system. This is due to the overlapping supports of such wavelets (which does not occur in the Haar case). Indeed, to compute a scaling coefficient $a_j(n)$ one needs not only the time-synchronous parents of $a_j(n)$ but also a number of neighboring coefficients depending on the supports of the analysis and synthesis wavelet. Thus, if we build a multiscale process where the states are defined as in the Haar case, i.e., $x_j(n) = \begin{bmatrix} a_j(n) & d_j(n) \end{bmatrix}^T$, but where we consider that the scaling and detail coefficients are computed using more regular wavelets, we will end up with a more complex graph structure of the scale-to-scale autoregression instead of a tree. This is illustrated in Figure 5.2 in the case of the Daubechies 4-tap filter [46–48].

In order to unify wavelets and MAR processes, the first issue, then, is to redefine the states so as to arrive at a tree dependency structure rather than a more complex graph. We will see that this can be done easily using state augmentation. The second and more difficult issue we must address is how to provide internality. These two issues will be the focus of the next section.

■ 5.2 MAR-Wavelet Processes

In this section, we first address the issue of defining the states of a MAR-wavelet process to obtain a tree dependency structure. We then address the issue of internality.

■ 5.2.1 Tree Structure for Synthesis

To see the intuition behind how to define the states in order to arrive at a tree structure, let us consider the simple case where h is the Daubechies 4-tap filter [46–48]. In this case, we have $\text{supp}(h) = [-1 : 2]$. Then, the wavelet reconstruction algorithm (5.8) implies that for every even integer n,

$$a_j(n-1) = \sum_{p=\frac{n}{2}-1}^{\frac{n}{2}} h(n-2p-1)a_{j-1}(p) + \sum_{p=\frac{n}{2}-1}^{\frac{n}{2}} g(n-2p-1)d_{j-1}(p)\,, \tag{5.11a}$$

$$a_j(n) = \sum_{p=\frac{n}{2}-1}^{\frac{n}{2}} h(n-2p)a_{j-1}(p) + \sum_{p=\frac{n}{2}-1}^{\frac{n}{2}} g(n-2p)d_{j-1}(p)\,, \tag{5.11b}$$

$$a_j(n+1) = \sum_{p=\frac{n}{2}}^{\frac{n}{2}+1} h(n-2p+1)a_{j-1}(p) + \sum_{p=\frac{n}{2}}^{\frac{n}{2}+1} g(n-2p+1)d_{j-1}(p)\,, \tag{5.11c}$$

$$a_j(n+2) = \sum_{p=\frac{n}{2}}^{\frac{n}{2}+1} h(n-2p+2)a_{j-1}(p) + \sum_{p=\frac{n}{2}}^{\frac{n}{2}+1} g(n-2p+2)d_{j-1}(p)\,. \tag{5.11d}$$

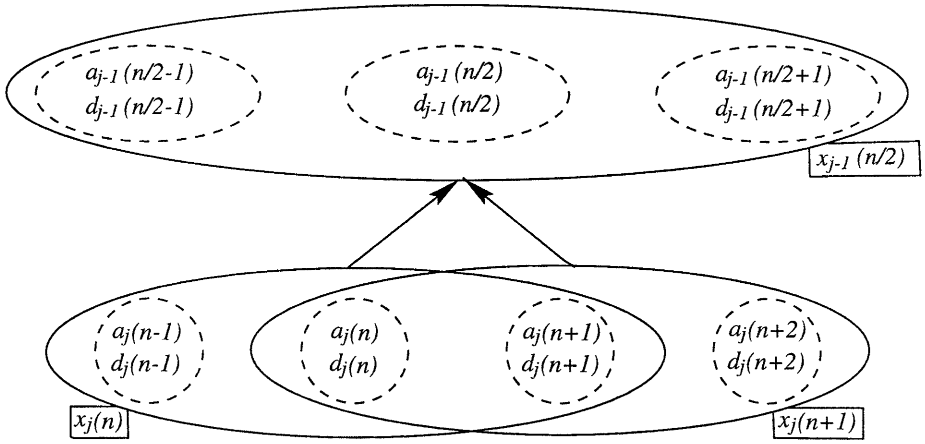


Figure 5.3. Through state augmentation, the dependency graph for the Daubechies 4-tap filter can be made into a tree. Here n is even.

Therefore, for every $j = 0, \ldots, M$ and for every $n = 0, \ldots, 2^j - 1$, if we choose each state $x_j(n)$ to be

$$x_j(n) = \begin{bmatrix} a_j(n-1) \\ a_j(n) \\ a_j(n+1) \\ d_j(n-1) \\ d_j(n) \\ d_j(n+1) \end{bmatrix} \tag{5.12}$$

it is clear from (5.11) that the scaling coefficients carried within each $x_j(n)$ depend only on the parent $x_{j-1}(\lfloor n/2 \rfloor)$ of $x_j(n)$ (see Figure 5.3).

In the general case (i.e., for any orthogonal or biorthogonal compactly supported wavelet), for every $j = 0, ..., M$ and for every $n = 0, ..., 2^j - 1$, the state at scale j and shift n is defined as

$$x_j(n) = \begin{cases} \begin{bmatrix} a_j(n - \widetilde{R} + 1) \\ \vdots \\ a_j(n + \widetilde{R} - 1) \\ d_j\big(n - \frac{\widetilde{R}+R}{2} + 1\big) \\ \vdots \\ d_j\big(n + \frac{\widetilde{R}+R}{2} - 1\big) \end{bmatrix} & \text{if } 0 \le j < M, \\ a_j(n) & \text{otherwise}. \end{cases} \tag{5.13}$$

The details showing that (5.13) implies that each state depends only on its parent, can be found in Appendix C. We then can show that

$$x_j(n) = \begin{cases} A_j(n)x_{j-1}\left(\lfloor n/2 \rfloor\right) + w_j(n) & \text{if } 0 \le j < M\,, \\ A_j(n)x_{j-1}\left(\lfloor n/2 \rfloor\right) & \text{if } j = M\,. \end{cases} \tag{5.14a}$$

where

$$w_j(n) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ d_j\left(n - \frac{\widetilde{R}+R}{2} + 1\right) \\ \vdots \\ d_j\left(n + \frac{\widetilde{R}+R}{2} - 1\right) \end{bmatrix} \tag{5.14b}$$

and where the first $2\widetilde{R} - 1$ entries of $w_j(n)$ are zero. The proof of (5.14a) and the expression for the matrices $A_j(n)$ can also be found in Appendix C. Assuming that $w(\cdot)$ is a white noise process uncorrelated with the root node state $x_0(0)$, (5.14a) represents a MAR process with dynamics matching the reconstruction algorithm associated with any compactly supported orthogonal or biorthogonal wavelet. In the sequel, we refer to this process as the *standard MAR-wavelet process.*

■ 5.2.2 Augmentation for Internality

If the coefficients $d_j(n)$ are considered as the detail coefficients computed using the wavelet decomposition algorithm and thus are deterministic inputs, then (5.14a) and (5.14b) is just a rewriting of the wavelet reconstruction algorithm (5.8). If, on the other hand, the coefficients $d_j(n)$ are generated as random variables, then (5.14a) and (5.14b) constitute a statistical *model* for a fine-scale process. However, almost surely, the states generated by this model do not consist of scaling and detail coefficients of the realized fine-scale process x^M. This is because the standard MAR-wavelet process is *not* internal. Indeed, as shown in Chapter 3, a necessary and sufficient condition for internality is that each state depends linearly on its immediate children. However, for each state $x_j(n)$, *only* $a_j(n)$ and $d_j(n)$ can be expressed as linear functions of the children states. From the wavelet decomposition algorithm, one can easily see that, when $\widetilde{R} > 1$, each state of the standard MAR-wavelet process is not a linear function of just its immediate children but of range of states at the next finer scale depending on the supports of the scaling functions.

The question now is how to build an internal MAR-wavelet process in order to ensure that the states consist of scaling and detail coefficients of the realized fine-scale process. This issue is, in fact, the one which seemed to doom the union between MAR processes and wavelets, and one of the contributions of this chapter is to solve this problem. This will be done by exhibiting and exploiting some relationships between scaling and detail coefficients and by appropriately modifying the state definition. We emphasize that this is purely deterministic analysis.

As explained above, the states of the standard MAR-wavelet process contain the information necessary for synthesis. To achieve internality, we need to augment each state so that the children of each state contain all the information necessary for *analysis.* Before showing how we augment the states, we need the following intermediate result which allows us to add only a few coefficients in the process of defining internal states.

Proposition 5.2.1. *There exist four matrices* L_1, J_1, L_2, J_2 *such that*

$$\begin{bmatrix} d_j\left(n - \frac{\widetilde{R}+R}{2} + 1\right) \\ \vdots \\ d_j(n-1) \end{bmatrix} = L_1 \begin{bmatrix} a_j(n - \widetilde{R} + 1) \\ \vdots \\ a_j\left(n - \frac{\widetilde{R}-R}{2} - 1\right) \end{bmatrix} + J_1 \begin{bmatrix} a_{j+1}(2n - \widetilde{R} + 1) \\ \vdots \\ a_{j+1}(2n + \widetilde{R} - 2) \end{bmatrix}, \tag{5.15a}$$

$$\begin{bmatrix} d_j(n+1) \\ \vdots \\ d_j\left(n + \frac{\widetilde{R}+R}{2} - 1\right) \end{bmatrix} = L_2 \begin{bmatrix} a_j\left(n + \frac{\widetilde{R}-R}{2} + 1\right) \\ \vdots \\ a_j(n + \widetilde{R} - 1) \end{bmatrix} + J_2 \begin{bmatrix} a_{j+1}(2n - \widetilde{R} + 3) \\ \vdots \\ a_{j+1}(2n + \widetilde{R}) \end{bmatrix}. \tag{5.15b}$$

Proof. See Appendix C. ■

Proposition 5.2.1 tells us that the detail coefficients that are contained in state $x_j(n)$ of the standard MAR-wavelet process (cf., (5.13)) can be expressed in terms of the scaling coefficients in $x_j(n)$ and the scaling coefficients of the child states of $x_j(n)$, namely $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$. As we now show, the significance of this fact is that, in order to achieve internality, we need only augment $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$ with a relatively small number of scaling coefficients and need not include any additional detail coefficients.

The idea behind constructing internal states is to define new states $x_j(n)$ in such a way so that the left child of $x_j(n)$ contains

$$\underline{a}_j^l \triangleq \begin{bmatrix} a_j(n - \widetilde{R} + 1) \\ \vdots \\ a_j(n - \frac{\widetilde{R}-R}{2} - 1) \end{bmatrix} \tag{5.16a}$$

and the right child of $x_j(n)$ contains

$$\underline{a}_j^r \triangleq \begin{bmatrix} a_j(n + \frac{\widetilde{R}-R}{2} + 1) \\ \vdots \\ a_j(n + \widetilde{R} - 1) \end{bmatrix}. \tag{5.16b}$$

However, having copied $\underline{a}_j^l$ and $\underline{a}_j^r$ from $x_j(n)$ to its children $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$, we must continue to pass $\underline{a}_j^l$ and $\underline{a}_j^r$ down to the children (and grand-children and so on) of $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$ to maintain internality. Of course we must do this for all j and n. This seems to suggest that the state dimensions will explode. However, by simply splitting at each step the necessary information between the two children, the state dimension remains bounded. The construction of the states is depicted in Figure 5.4 in the simple case of Daubechies 4-tap filter. To define rigorously the internal states in the general case, we define recursively the sequence of partitioned vectors:[3]

$$\xi_0(0) \triangleq \left[\begin{array}{c} \xi_0^1(0) \\ \hdashline \xi_0^2(0) \end{array}\right] \triangleq \left[\begin{array}{c} a_0\big(-\widetilde{R}+1\big) \\ \vdots \\ a_0\big(-\frac{\widetilde{R}-R}{2}-1\big) \\ \hdashline a_0\big(\frac{\widetilde{R}-R}{2}+1\big) \\ \vdots \\ a_0\big(\widetilde{R}-1\big) \end{array}\right] \tag{5.17a}$$

and for $j = 1, \dots, M$ and $n = 0, \dots, 2^j - 1$,

$$\xi_j(n) \triangleq \left[\begin{array}{c} \xi_j^1(n) \\ \hdashline \xi_j^2(n) \end{array}\right] \triangleq \begin{cases} \left[\begin{array}{c} \xi_{j-1}^1(n/2) \\ \hdashline a_{j-1}\big(n/2-\widetilde{R}+1\big) \\ \vdots \\ a_{j-1}\big(n/2-\frac{\widetilde{R}-R}{2}-1\big) \end{array}\right] & \text{if } n \text{ is even}, \\ \left[\begin{array}{c} \xi_{j-1}^2(\lfloor n/2 \rfloor) \\ \hdashline a_{j-1}\big(\lfloor n/2 \rfloor+\frac{\widetilde{R}-R}{2}+1\big) \\ \vdots \\ a_{j-1}\big(\lfloor n/2 \rfloor+\widetilde{R}-1\big) \end{array}\right] & \text{if } n \text{ is odd}. \end{cases} \tag{5.17b}$$

We then have the final result in which we have augmented the states corresponding to the standard MAR-wavelet model in order to achieve internality.

[3] The notation $\left[\begin{array}{c} a \\ \hdashline b \end{array}\right] = \left[\begin{array}{c} c \\ \hdashline d \end{array}\right]$ means that $a = c$ and $b = d$.

Proposition 5.2.2. *The MAR process for which the states are defined by*

$$x_j(n) = \begin{cases} \begin{bmatrix} a_0(-\widetilde{R}+1) \\ \vdots \\ a_0(\widetilde{R}-1) \\ d_0(-\frac{\widetilde{R}+R}{2}+1) \\ \vdots \\ d_0(\frac{\widetilde{R}+R}{2}-1) \end{bmatrix} & \text{if } j = 0, \\ \begin{bmatrix} a_1(n-\widetilde{R}+1) \\ \vdots \\ a_1(n+\widetilde{R}-1) \\ d_1(n-\frac{\widetilde{R}+R}{2}+1) \\ \vdots \\ d_1(n+\frac{\widetilde{R}+R}{2}-1) \\ \xi_0^{n+1}(0) \end{bmatrix} & \text{if } j = 1, \\ \begin{bmatrix} a_j(n-\widetilde{R}+1) \\ \vdots \\ a_j(n+\widetilde{R}-1) \\ d_j(n-\frac{\widetilde{R}+R}{2}+1) \\ \vdots \\ d_j(n+\frac{\widetilde{R}+R}{2}-1) \\ \xi_j(n) \end{bmatrix} & \text{otherwise} \end{cases} \tag{5.18}$$

is internal.

Proof. See Appendix C. ■

We refer to this new process as the *internal MAR-wavelet process.* Notice that the size of each $\xi_j(n)$ is $\widetilde{R} + R - 2$. Thus the maximal state dimension of the internal MAR-wavelet process is $4\widetilde{R} + 2R - 4$. With Proposition 5.2.2, we have shown how to build internal MAR processes based on any compactly supported orthogonal or biorthogonal wavelet. For future reference, we point out that Proposition 5.2.1 and the wavelet analysis equation (5.7a) implicitly define the local internal matrices $\{V_j(n)\}$ each of which relates a state to its children states. Moreover, each state depends only on the scaling coefficients of its children states and not on the detail coefficients. This completes our unification of wavelets with MAR processes.

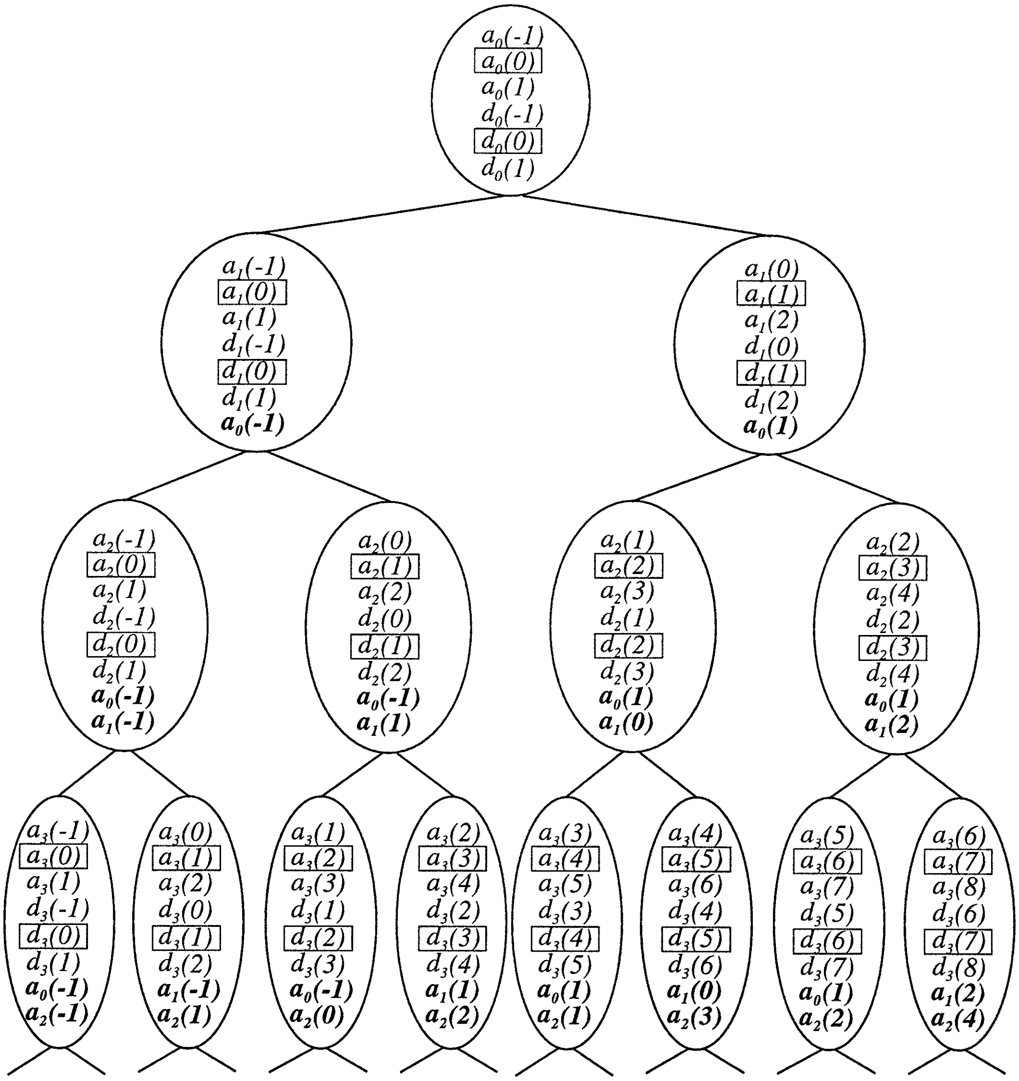


Figure 5.4. Example of the internal MAR-wavelet process with the Daubechies 4-tap filter. Scaling coefficients in bold illustrate the necessary information transmitted from one scale to the next. The boxed coefficients are a linear function of the coefficients of their children by virtue of the wavelet decomposition algorithm.

■ 5.3 MAR-Wavelet Models

In this section, we apply the MAR-wavelet processes developed in the previous sections to the problem of modeling a fine-scale random signal, f^M. The standard MAR-wavelet process, defined by (5.14a) and (5.14b), can be used as an approximate model for a stochastic process by assuming that the detail coefficients are white noise. We will call this model the *standard MAR-wavelet model.* However, the states realized using this model are not consistent with the fine-scale realized process x^M in the sense that these states do not represent, with probability one, scaling and detail coefficients of x^M. This is because of the lack of internality, as discussed in Section 5.2.2. Notice that the assumption of the whiteness of $w(\cdot)$ (defined by (5.14b)) is an approximation if $\widetilde{R} > 1$. Indeed, for n and m such that $0 < |n - m| \leq R + \widetilde{R} - 2$, it is clear that, at a given scale j, $w_j(n)$ and $w_j(m)$ are correlated since they share at least one detail coefficient.

By achieving internality, the states of the internal MAR-wavelet process (5.18) are forced to be consistent with the fine-scale realized process. We can use the internal MAR-wavelet process to build what we shall call the *internal MAR-wavelet model* to approximate the given statistics of a fine-scale process. Given these fine-scale statistics and using internality, the statistics of any MAR state and the statistics between each state and its parent are readily computed as we have seen in previous chapters. As a result, we can immediately define the linear dynamics of a MAR model, dynamics that incorporate optimal prediction from parent to child. The prediction errors are then modeled as white driving noise in order to satisfy the Markov property.

The implications of this are twofold. First, the resulting internal model, in general, produces fine-scale statistics that only approximate the desired ones (because of our insistence on modeling the coarse-to-fine prediction errors as white). To be sure, our internal MAR-wavelet model *does* produce the correct marginal statistics at each node and the correct joint statistics for each state and its parent, but other statistics (e.g., cross-covariance for two nodes at the same scale) are only captured approximately. The second point is that the coarse-to-fine dynamics so defined are in general *very* different from standard wavelet modeling. In particular, these dynamics exploit correlation between detail coefficients *and* coarser scale scaling and detail coefficients by performing optimal prediction and then assuming that only the errors in these predictions are white. This is in marked contrast to one common approach in using wavelets for modeling stochastic processes in which the detail coefficients are themselves modeled as white (i.e., the wavelet representation is *assumed* to be the Karhunen-Loeve decomposition). In our case, since we allow MAR dynamics, we do not *need* to have K-L diagonalization. Rather, the success of our method in approximating stochastic processes relies only on the weaker requirement that the *errors* in predicting finer scale detail coefficients from coarser scale coefficients are white. As we illustrate, an implication of this is that we can use fairly short wavelets, implying lower state dimensions, which certainly do *not* do a good job of whitening the details (as evidenced by our results using the non-internal standard MAR-wavelet models), but which do remarkably well for our internal models.

Using the optimal prediction procedure of the internal MAR-wavelet model, we thus incorporate a synthesis algorithm for the detail coefficients themselves, in addition to the usual wavelet reconstruction algorithm for the scaling coefficients. The initialization for this new synthesis algorithm is given by the statistics of the scaling and detail coefficients contained in the root node state, $P_{x_0(0)} = W_0(0)P_{f^M}W_0(0)^T$. To specify the dynamics for the detail coefficients which complement the usual synthesis algorithm for the scaling coefficients (cf., (5.8)), let $\underline{d}_j(n)$ represent the detail coefficients carried by the state $x_j(n)$ defined by (5.18), then

$$\underline{d}_j(n) = P_{\underline{d}_j(n)x_{j-1}(\lfloor n/2 \rfloor)} P^{-1}_{x_{j-1}(\lfloor n/2 \rfloor)} x_{j-1}(\lfloor n/2 \rfloor) + \widetilde{w}_j(n) \tag{5.19}$$

where the covariance matrix for $\widetilde{w}_j(n)$ is

$$P_{\widetilde{w}_j(n)} = P_{\underline{d}_j(n)} - P_{\underline{d}_j(n)x_{j-1}(\lfloor n/2 \rfloor)} P^{-1}_{x_{j-1}(\lfloor n/2 \rfloor)} P^T_{\underline{d}_j(n)x_{j-1}(\lfloor n/2 \rfloor)} \,. \tag{5.20}$$

The matrices $P_{\underline{d}_j(n)}$, $P_{\underline{d}_j(n)x_{j-1}(\lfloor n/2 \rfloor)}$, and $P_{x_{j-1}(\lfloor n/2 \rfloor)}$ are submatrices of $P_{x_j(n)}$ and $P_{x_j(n)x_{j-1}(\lfloor n/2 \rfloor)}$, the state covariance and child-parent cross-covariance. These are computed, as described in previous chapters, via

$$P_{x_j(n)} = W_j(n)P_{f^M}W_j(n)^T \,, \tag{5.21a}$$

$$P_{x_j(n)x_{j-1}(\lfloor n/2 \rfloor)} = W_j(n)P_{f^M}W_{j-1}(\lfloor n/2 \rfloor)^T \tag{5.21b}$$

where the internal matrices $W_j(n)$ are implicitly given by Proposition 5.2.2. Note that only the detail coefficients have a driving noise component and the dynamics of the scaling coefficients are deterministic. This property is what ensures that our models are, in fact, internal. Indeed, as discussed at the end of Section 5.2, the local internal matrices, $V_j(n)$, for an internal MAR-wavelet process only act on scaling coefficients. The detail coefficients are in the null-space of $V_j(n)$. Since the driving noise is in the null-space of $V_j(n)$, the relationship

$$x_j(n) = V_j(n) \begin{bmatrix} x_{j+1}(2n) \\ x_{j+1}(2n+1) \end{bmatrix} \tag{5.22}$$

is deterministic and the model is internal (recall our discussion of this point in Section 4.1.2).

The prediction errors $\widetilde{w}_j(n)$ are not white in general. This can be easily seen from the fact that the states of the internal MAR-wavelet model contain duplicated detail coefficients. Yet, we assume that the prediction errors $\widetilde{w}_j(n)$ in (5.19) are white noise so that we arrive at a MAR model. This internal MAR-wavelet model is, therefore, approximate. Note that an advantage of the internal MAR-wavelet model and, in fact, of any internal model is that it achieves the correct variances (i.e., the diagonal elements of P_{x^M} match *exactly* those of P_{f^M}).

We emphasize another important point. In real world problems, the user may not know how to choose the appropriate wavelet which will do a good job in decorrelating the process under study. Thus, the resulting detail and scaling coefficients may be strongly correlated. In this case, our internal MAR-wavelet model based on optimal prediction from parent to child may still be quite accurate because it exploits these potential correlations, as well as those between detail and scaling coefficients. We will illustrate this later in our examples.

In Section 2.3.1 we compared the standard MAR-Haar model with the internal one for modeling fBm. Recall that with the Haar wavelet, the detail coefficients which are not neighbors (in space and scale) are in general strongly correlated. Therefore, even with the optimal prediction procedure of the internal MAR-Haar model the resulting realized covariance matrix is quite crude (see Figure 2.3) since it captures only the correlations between a detail coefficient at a given scale and the time-synchronous detail and scaling coefficients at the previous coarser scale.

One way to overcome the limitations of the Haar wavelet is to build an internal MAR-wavelet model using an analyzing wavelet with a large number of vanishing moments. With such a wavelet, the detail coefficients which are not neighbors in space and scale will, in general, be better decorrelated and the potential correlations will reside only between neighboring coefficients. Then, our optimal prediction procedure will exploit these residual correlations between detail and scaling coefficients and do the best job in linearly predicting the detail coefficients.

However, this is not the only solution. One can still build accurate models without necessarily using an analyzing wavelet with large number of vanishing moments. Indeed, all we need in order to have accurate models is to provide a good approximation to the Markov property. Therefore, accurate models will be provided using *any* wavelet yielding scaling and detail coefficients such that the states they form approximatively fulfill the *conditional* decorrelation role of the Markov property.

5.3.1 Realized Covariance Examples

In this section, we apply our MAR-wavelet models to approximate the statistics of fBm using different wavelets. Figure 5.5(a) and Figure 5.5(b) show P_{fM} for 64 samples of fBm(0.3) and fBm(0.7), respectively, on the interval $(0, 1]$. For purposes of comparison, Figure 5.5(c) and Figure 5.5(d) illustrate $|P_{fM} - P_{xM}|$ where P_{xM} has been computed using the scale-recursive, predictive efficiency realization method of Section 4.1 (with state dimension 8) and P_{fM} is from Figure 5.5(a) and Figure 5.5(b), respectively. We will compare internal MAR-wavelet models to these predictive efficiency models and to standard MAR-wavelet models which assume the whiteness of the detail coefficients. We use the Daubechies orthogonal wavelet with 2 vanishing moments (Daub4), the Daubechies orthogonal wavelet with 3 vanishing moments (Daub6), the spline biorthogonal wavelet (Spline13) such that[4] $\widetilde{H}(z)$ (respectively $H(z)$) has 3 (respectively 1) zeros at $z = -1$, and the spline biorthogonal wavelet (Spline31) such that $\widetilde{H}(z)$ (respectively $H(z)$) has 1 (respectively 3) zeros at $z = -1$.

Figure 5.6(a), Figure 5.6(b), and Figure 5.6(c) display the element-wise absolute

[4] $H(z)$ (respectively $\widetilde{H}(z)$) is the z-transform [144, 145] of $h(n)$ (respectively $\widetilde{h}(n)$).

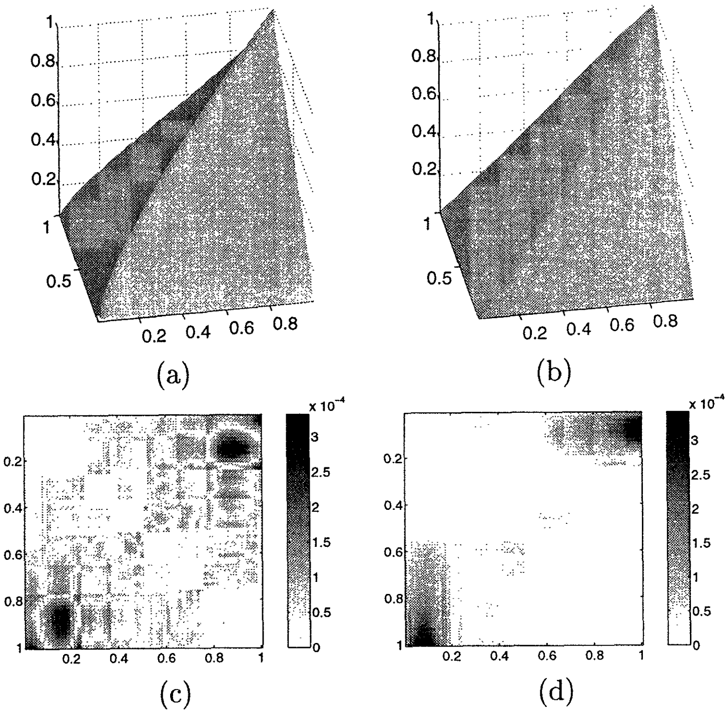


Figure 5.5. (a) Exact covariance matrix P_{fM} for 64 samples of fBm(0.3) on $(0, 1]$. (b) Same as (a) but for fBm(0.7). (c) $|P_{fM} - P_{xM}|$ where P_{xM} is based on the method of Section 4.1 (state dimension 8) and P_{fM} is from (a). (d) Same as (c) but P_{fM} is from (b).

value of the difference between P_{fM} and P_{xM} obtained by the standard MAR-wavelet model for an fBm(0.3) using, respectively, Daub4, Daub6, and Spline13. The improvement with respect to the standard MAR-Haar model of Section 2.3.1 is clear, as expected, since we are using analyzing wavelets with more than 1 vanishing moment. However, the approximation is not satisfactory which is not surprising since the detail coefficients are not exactly decorrelated using these wavelets. Note that Daub6 does better than Spline13 because Daub6 is an orthogonal wavelet and is smoother than the analyzing wavelet of Spline13.

Now, with the internal MAR-wavelet model, the detail coefficients are no longer assumed to be white noise. Instead, they are computed using the optimal prediction procedure described previously. Therefore, the internal MAR-wavelet model will better approximate the statistics of fBm. Figure 5.7(a), Figure 5.7(b), and Figure 5.7(c) display the element-wise absolute value of the difference between P_{fM} and P_{xM} for fBm(0.3) using, respectively, Daub4, Daub6, and Spline13. The improvement with respect to the standard MAR-wavelet model is clear. Figure 5.8(a), Figure 5.8(b), and Figure 5.8(c), display the same element-wise absolute value of the difference obtained using the internal MAR-wavelet model for fBm(0.7). Note that the internal MAR-wavelet models do not perform as well as the internal models based on predictive efficiency developed in Section 4.1, even in cases for which the internal MAR-wavelet model has a higher state dimension than its predictive efficiency counterpart. This can be seen by comparing the images in Figure 5.7 and Figure 5.8 with Figure 5.5(c) and Figure 5.5(d), respectively. All of the internal MAR-wavelet models considered in these figures produce errors two or more orders of magnitude larger than the predictive efficiency models. This fact is not surprising considering that the predictive efficiency method is based on selecting internal matrices to provide optimal conditional decorrelation given a state dimension constraint. In contrast, the MAR-wavelet method avoids solving optimization problems by simply selecting the internal matrices from a wavelet basis.

To illustrate, in the case of fBm, the fact that even with relatively non-regular wavelets, our internal MAR-wavelet model can provide very accurate models, we use the biorthogonal wavelet Spline31. The analyzing wavelet for Spline31 has only 1 vanishing moment and the synthesis wavelet is extremely singular (see Figure 5.9). Figure 5.10(a) displays the element-wise absolute value of the difference between P_{fM} and P_{xM} using the standard MAR-wavelet model. One sees that the approximation is extremely bad, which is not surprising given the properties of Spline31 and the weakness of the assumption that the detail coefficients are white. However, using the internal MAR-wavelet model, the approximation is very accurate as displayed in Figure 5.10(b). Furthermore, notice that this approximation is more accurate than the one illustrated in Figure 5.8(c) in which the state dimension is larger. Indeed, in Figure 5.10(b) we have $R = \widetilde{R} = 2$ and thus the maximum state dimension is 8 while in Figure 5.8(c) we have $R = \widetilde{R} = 3$ and thus the maximum state dimension is 10. This shows the power of the optimal prediction procedure in approximating the Markov property even without considering analyzing wavelets with a large number of vanishing moments. Not surprisingly, however, the errors of Figure 5.10(b) are orders of magnitude larger than those of the predictive efficiency model of Figure 5.5(d).

■ 5.3.2 Sample-Path Generation and Estimation Examples

Next, we use the fast signal processing algorithms associated with the MAR framework to synthesize fBm sample-paths and to perform estimation from incomplete measurements corrupted by nonstationary noise. Figure 5.11(a) and Figure 5.11(b) display 256-point sample-paths using the internal MAR-wavelet model with Daub6 for fBm(0.3) and fBm(0.7), respectively. Figure 5.12(a) displays an exact 64-point realization of fBm(0.3). Figure 5.12(b) displays noisy observations of Figure 5.12(a) where observations are only available on $(0, 1/3]$ (over which the white measurement noise has variance 0.3) and $(2/3, 1]$ (over which the white measurement noise has variance 0.5). Figure 5.12(c) displays the MAR estimates based on Figure 5.12(b) using the internal wavelet model with Daub6. The MAR estimates are the solid line and the optimal estimates based on the exact statistics are the dash-dot line. The plus/minus one standard deviation error bars are the dashed line.

Figure 5.12(d)–Figure 5.12(f) illustrates the same processing but for fBm(0.7). No-

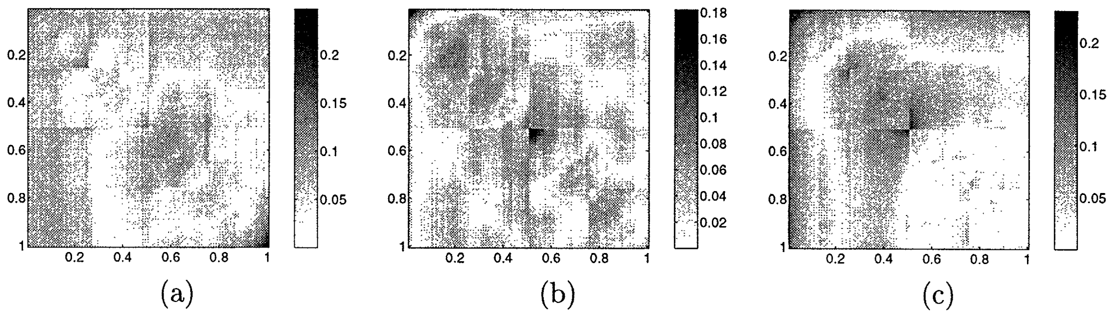


Figure 5.6. $|P_{fM} - P_{xM}|$ for fBm(0.3) using the standard MAR-wavelet model. (a) Daub4 (state dimension 6). (b) Daub6 (state dimension 10). (c) Spline13 (state dimension 8).

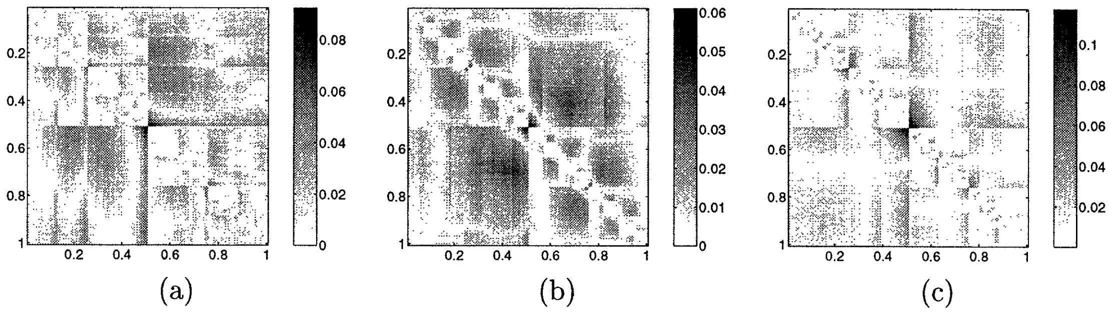


Figure 5.7. $|P_{fM} - P_{xM}|$ for fBm(0.3) using the internal MAR-wavelet model. (a) Daub4 (state dimension 8). (b) Daub6 (state dimension 14). (c) Spline13 (state dimension 10).

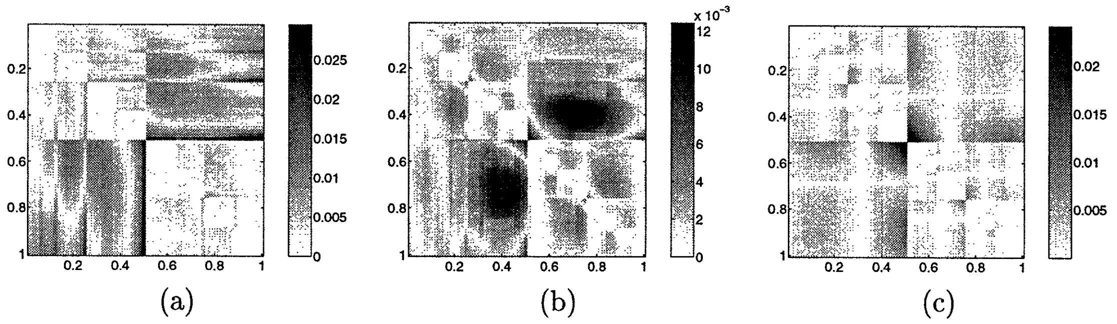


Figure 5.8. $|P_{fM} - P_{xM}|$ for fBm(0.7) using the internal wavelet model. (a) Daub4 (state dimension 8). (b) Daub6 (state dimension 14). (c) Spline13 (state dimension 10).

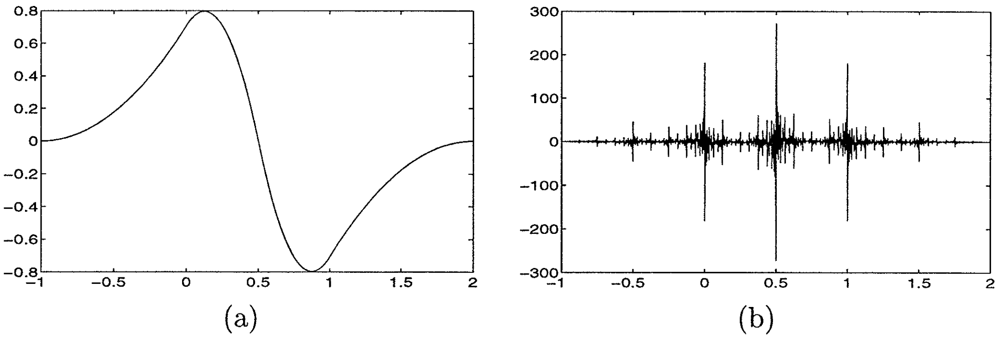


Figure 5.9. Spline biorthogonal wavelet (Spline31). (a) Analyzing wavelet. (b) Synthesis wavelet.

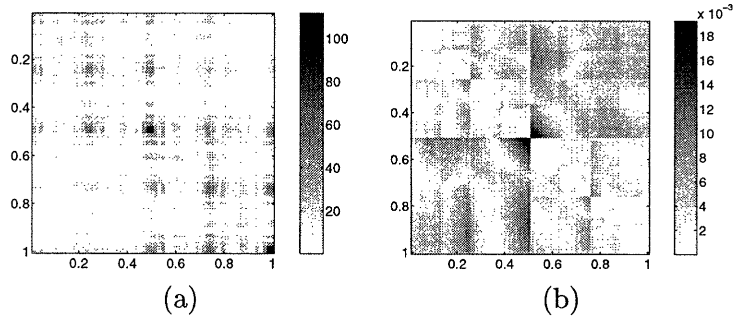


Figure 5.10. $|P_{fM} - P_{xM}|$ for fBm(0.7) using (a) the standard MAR-wavelet model with Spline31 (state dimension 6) and (b) the internal MAR-wavelet model with Spline31 (state dimension 8).

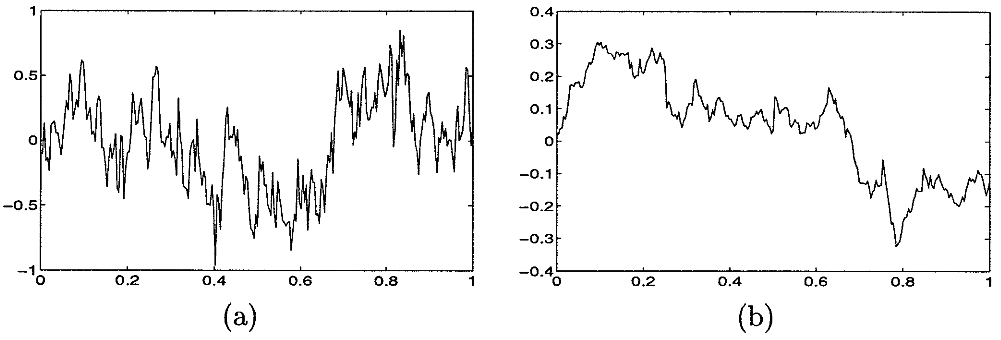


Figure 5.11. Sample-paths using the internal MAR-wavelet model with Daub6. (a) fBm(0.3). (b) fBm(0.7).

tice that in both Figure 5.12(c) and in Figure 5.12(f) the optimal estimate based on the exact statistics is not easily distinguishable from the MAR estimate since the two nearly coincide. Also, the estimation error standard deviations that the MAR estimator provides are very close to the ones based on the exact statistics (although we have not plotted the latter in our examples). More importantly, the difference between the optimal estimate and the MAR estimate is well within the one standard deviation error bars. This demonstrates that the degree to which our internal MAR-wavelet model deviates from the exact model is statistically irrelevant.

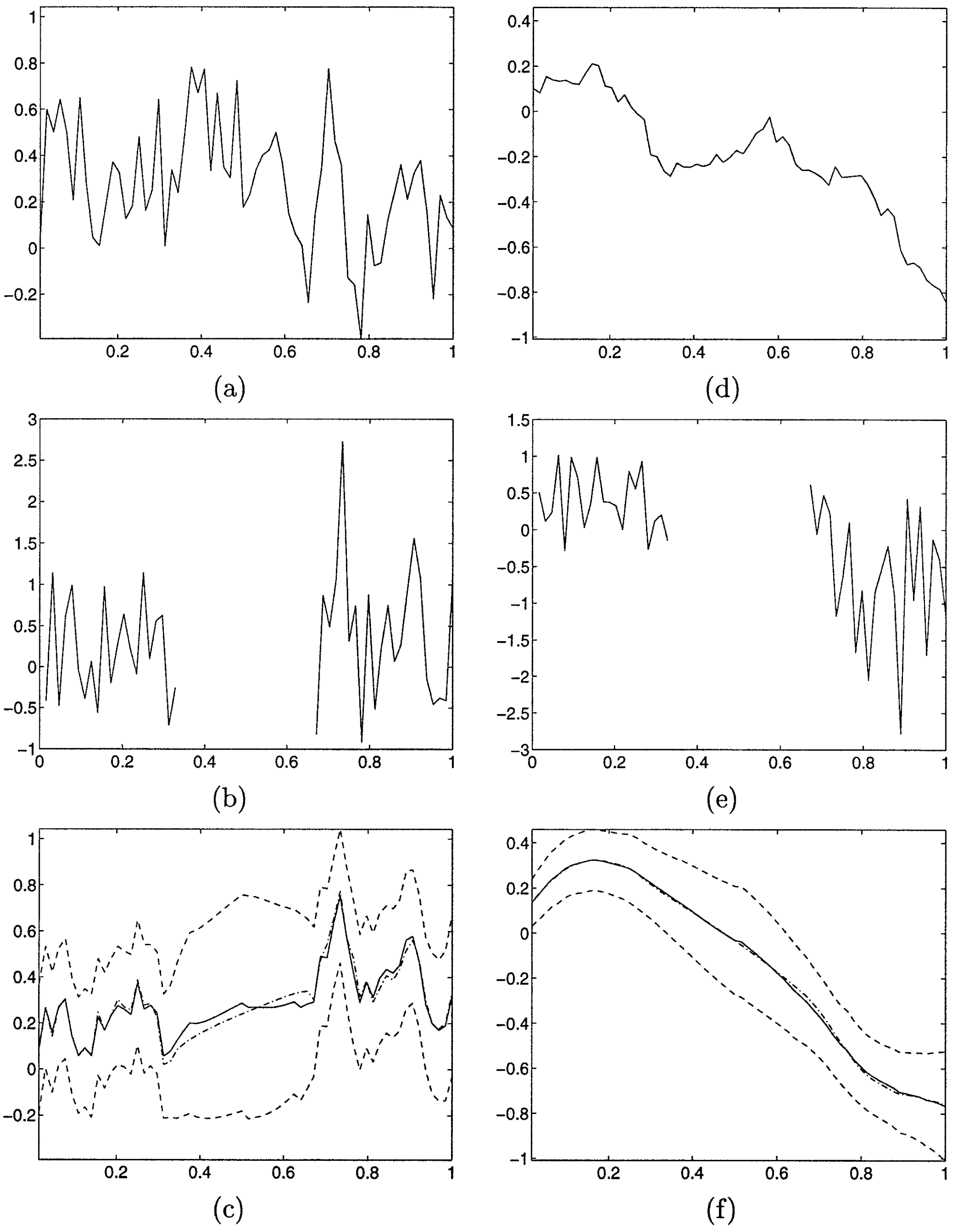


Figure 5.12. MAR estimation of fBm(0.3) using the internal MAR-wavelet model with Daub6. (a) Sample-path using exact statistics. (b) Noisy, incomplete observations of (a). The noise variance over $(0, 1/3]$ is 0.3 and over $(2/3, 1]$ is 0.5. (c) MAR estimates are the solid line and optimal estimates based on the exact statistics are the dash-dot line. The plus/minus one standard deviation error bars are the dashed line. (d)–(f) differ from (a)–(c) only in that they are based on fBm(0.7) rather than fBm(0.3).

Chapter 6

Covariance Extension

THIS chapter addresses a major limitation of *all* previously-developed systematic MAR model identification methods (including those presented in the preceding chapters of this thesis). This limitation is the requirement of *complete and precise* knowledge of *every* entry of P_{fM}, the covariance matrix of the signal being modeled. For large problems involving hundreds of thousands or millions of variables, such as those that arise in geophysical or remote sensing applications [60, 99, 138, 139, 143, 194], it is unreasonable to assume that one will have complete knowledge of the statistics of the underlying processes. Further, even in cases for which every element of P_{fM} is known, such knowledge is practically useless because, unless structure can be exploited, the amount of memory required to store P_{fM} is prohibitive, even for image processing problems of modest size.[1] Moreover, elements of P_{fM} about which information is available are unlikely to be specified with precision. Rather, bounds are likely to be given.

If the MAR framework is ultimately to be used to address large image processing problems with the features just described, there is considerable motivation to develop model identification techniques that do not require complete and precise knowledge of all of P_{fM}. In this chapter we take a first step toward this goal. In particular, while we still require that elements of P_{fM} are precisely specified (i.e., we cannot accommodate bounds), we relax the assumption that *every* element is known. We then address the problem of building a MAR model that is consistent with the known elements of P_{fM} and which implicitly provides reasonable values for the unknown ones (i.e., the realized covariance matrix must be a valid one and so must be symmetric and positive-definite).

The problem of inferring unknown covariance elements from known ones is the *covariance extension* problem which has been extensively studied in recent years. In particular, much has been written about *maximum-entropy* covariance extension, its applications in spectral estimation [22, 23, 101, 147, 171] and VLSI modeling [55], its connection with autoregressive (all-pole) models [21, 86, 95, 120, 171], and its link to lattice structures for FIR filters [81, 89, 117]. We will introduce the maximum-entropy covariance extension problem in Section 6.1 and indicate in what sense this chapter

[1] Memory requirements for a covariance matrix associated with a 256 × 256 image are on the order of tens of gigabytes.

addresses it. Many of the results on covariance extension are most naturally expressed in graph-theoretic terms and we provide, in Section 6.2, a brief review of the relevant graph-theoretic concepts and results. Following this, we review some of the previously-developed theory regarding covariance extension in Section 6.3. In Section 6.4 we build upon the known results by providing a Levinson-like algorithm which may be applied to the computation of covariance extensions under very general conditions. Our ultimate interest is not in extending a partially specified covariance matrix to a fully specified one but, rather, in directly building a MAR *model* for such an extension (without explicitly calculating every or even most of the unknown elements of P_{fM}). This is the topic of Section 6.5 in which we consider MAR models for the maximum-entropy extension of covariance matrices that are specified on diagonal bands including and adjacent to the main diagonal. An important result which we show is that building a MAR model for an extension of this type can be done with *vastly* fewer computations than finding the full covariance extension explicitly. Moreover, the computational complexity of building a MAR model is comparable to that which is obtained by Levinson's algorithm in designing an autoregressive (AR) time-series model.

6.1 Covariance Extension and Completion

In this section we introduce the covariance extension and completion problems and discuss the classical Levinson algorithm. The covariance extension and completion problems begin with a *partial covariance matrix* which we define shortly. Let $V = \{0, 1, \ldots, N-1\}$ and let E be a subset of $V \times V$. Then a *partial matrix*, P_E, is the set[2]

$$P_E \triangleq \{(i, j, p_{i,j}) \mid (i,j) \in E\}. \tag{6.1}$$

We shall call E the *support set* for P_E. One can also think of P_E as a partially filled matrix where

$$P_E(i,j) = \begin{cases} p_{i,j} & \text{if } (i,j) \in E\,, \\ ? & \text{otherwise}\,. \end{cases} \tag{6.2}$$

Hereafter, we will only consider support sets that are *symmetric.* By a symmetric support set we mean that $(i,j) \in E$ if and only if $(j,i) \in E$. Additionally, in order to avoid pathologies (to be discussed), we assume that E contains all pairs of the form (i,i). Notice that if the elements of P_E are drawn from a positive-definite covariance matrix, then every maximal principal minor[3] of P_E is positive definite, a property we denote by $P_E > 0$.

[2] In other chapters the notation P_E denotes a covariance matrix for random vector E. In this chapter E is *not* a random vector, and we define objects of the form P_E differently as discussed in the text.

[3] A *principal minor* of P_E is a subset (which we view as a matrix) of P_E given by $\{P_E(i,j)\}_{(i,j)\in\alpha\times\alpha}$ where $\alpha \times \alpha \subset E$. A principal minor is *maximal* if it is not a subset (submatrix) of any other principal minor.

Definition 6.1.1 (Partial Covariance Matrix). *P_E is a* partial covariance matrix *if it is a partial matrix (i.e., it is a set of the form given by* (6.1)*) with $P_E(i,j) = P_E(j,i)$ and $P_E > 0$.*

Given a partial covariance matrix P_E, the *covariance extension* problem is to find another partial covariance matrix P_F where $E \subset F$ and $P_F \subset P_E$ so that P_F agrees with P_E on the index set E. A partial covariance matrix P_F that satisfies these criteria is called an *extension* of P_E. A *covariance completion* is a covariance extension with $F = V \times V$. That is, the covariance completion problem is to find a fully-specified valid covariance matrix $P > 0$ that agrees with P_E on the set E. If any completion of P_E exists then there exists a unique completion with maximal entropy which also coincides with the completion with maximal determinant [88, 120]. There are two additional points worth emphasizing. First, $P_E > 0$ is a necessary condition for the existence of extensions and completions. As we shall see, however, it is not a sufficient condition. Second, note that the diagonal elements of the partial covariance matrix P_E (i.e., $P_E(i,i)$ for $i \in V$) are defined because E contains all pairs (i,i). This is a necessary, but not sufficient, condition for the existence of a maximum-entropy completion. Indeed, if it were otherwise the case then the determinant (and, hence, the entropy) of the completion could be made arbitrarily large by increasing the value of any unspecified diagonal element.

The maximum-entropy completion of P_E, denoted by P_{ME}, can be characterized by the pattern of zeros in its inverse. In particular, $P_{\text{ME}}^{-1}(i,j) = 0$ if $(i,j) \notin E$ (i.e., $P_E(i,j)$ is undefined). This fact follows from the well-known solution to the problem of finding the entropy-maximizing probability density function given moment constraints [34,120]. In this case, the entropy-maximizing density has the functional form

$$C \exp\left\{ \sum_{(i,j)\in E} \lambda_{i,j} z_i z_j \right\} \tag{6.3}$$

where C is a normalization constant, $\lambda_{i,j}$ is a Lagrange multiplier, and $P_E(i,j) = \text{cov}(z_i, z_j)$ are the given constraints. The form of the density of (6.3) is Gaussian (i.e., it has the form $C\exp(-\frac{1}{2}z^T P^{-1} z)$) and clearly indicates that $P_{\text{ME}}^{-1}(i,j) = 0$ for $(i,j) \notin E$.

In this chapter our ultimate goal is not to compute P_{ME} but, rather, to design a MAR model for it. As we have noted in previous chapters, often an *implicit* representation of a covariance matrix is of greater practical utility than an *explicit* one. Implicit representations in the form of parameterized stochastic models often have structure that can be exploited to achieve efficiencies in computation and storage. Two such representations have been discussed in this thesis: MAR and state-space models. The well-known connection between AR processes[4] and maximum-entropy covariance com-

[4] An AR process can be written in state-space form as discussed in Section 2.3.2. However, not every state-space process is an AR process.

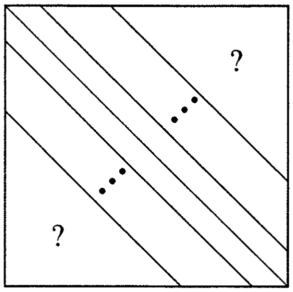


Figure 6.1. P_E where E is given by (6.4).

pletion will be discussed shortly. The connection to MAR processes has, until now, been unknown and is the topic of Section 6.5.

To make the connection between AR models and maximum-entropy covariance completion, consider the classical and simple case for which P_E is specified on $2k+1$ diagonal bands that include and are adjacent to the main diagonal (see Figure 6.1). That is, the support set E is

$$E \triangleq \{(m,n) \mid |m-n| \leq k\}. \tag{6.4}$$

Hence, P_{ME}^{-1}, the inverse of the maximum-entropy extension of P_E, has zeros off the $2k+1$ diagonal bands specified by E. This fact and the correspondence between inverse-covariance zeros and conditional decorrelation [45,170] implies that P_{ME} is a covariance matrix for a k-th order Markov process. Since any k-th order Markov process can be written as a k-th order AR (all-pole) process, there exists a state-space model for P_{ME} with state dimension k (as discussed in Section 2.3.2). For the case for which the known diagonals of P_E are constant-valued[5] so that $p_{i,j} = p_{i-j}$, the autoregression has the form

$$z(n) = a_1^k z(n-1) + a_2^k z(n-2) + \cdots + a_k^k z(n-k) + \mu(n) \tag{6.5}$$

where $\mu(\cdot)$ is white and $\begin{bmatrix} z(0) & z(1) & \cdots & z(N-1) \end{bmatrix}^T$ has covariance matrix P_{ME}. The superscript k of a_i^k reminds us that the autoregression parameters are associated with a k-th order model.

We now turn to the problem of computing the parameters of (6.5), a stationary AR process corresponding to P_{ME}, using Levinson's algorithm [21,89,171]. Levinson's algorithm, in its most basic form, can be used to solve a Toeplitz linear system that arises in LLS estimation[6] order-recursively by first considering a small linear system and then sequentially increasing the system's size. This procedure can be harnessed

[5] This corresponds to the assumption that the known elements of P_E correspond to a stationary process. We relax this assumption of stationarity later.

[6] A Toeplitz linear system is a system of linear equations $Ax = b$ in which A has constant-valued diagonals (i.e., is a Toeplitz matrix).

to compute the parameters of (6.5) order-recursively with complexity $O(k^2)$ as follows. The first-order LLS prediction of $z(n)$ based on $z(n-1)$ has the form

$$\widehat{z}^1(n) = a_1^1 z(n-1) \tag{6.6a}$$

where

$$a_1^1 = \frac{p_1}{p_0}. \tag{6.6b}$$

The superscript of $\widehat{z}^1(n)$ and a_1^1 reminds us that this is a first-order prediction. This notation is necessary because, for $j \neq 1$, $\widehat{z}^1(n) \neq \widehat{z}^j(n)$ and $a_1^1 \neq a_1^j$ where a_1^j is the first AR coefficient associated with the j-th order LLS prediction $\widehat{z}^j(n)$. For reasons that will become clear shortly, we define $\rho_1 \triangleq a_1^1$.

Continuing, the second-order LLS prediction has the form

$$\widehat{z}^2(n) = a_1^2 z(n-1) + a_2^2 z(n-2). \tag{6.7}$$

To obtain a recursion, we need to relate (6.7) to (6.6a). This relation is achieved by first decorrelating $z(n-2)$ from $z(n-1)$ using Gram-Schmidt orthogonalization. Doing so results in

$$\widehat{z}^2(n) = \underbrace{\rho_1 z(n-1)}_{\widehat{z}^1(n)} + \rho_2 \underbrace{\Big(z(n-2) - \widehat{\mathrm{E}}\big[z(n-2) \mid z(n-1)\big]\Big)}_{b_1} \tag{6.8a}$$

where

$$\rho_2 = \frac{\mathrm{E}[z(n)b_1]}{\mathrm{E}[b_1]^2}. \tag{6.8b}$$

There are two important things to notice about (6.8a). First, it is a sum of $\widehat{z}^1(n)$ and a term orthogonal to $\widehat{z}^1(n)$ which we have denoted $\rho_2 b_1$. Second, $\widehat{\mathrm{E}}\big[z(n-2) \mid z(n-1)\big]$ is a one-step *backward* temporal prediction. Due to the assumption of stationarity, this one-step backward prediction has the same form as the one-step forward prediction of (6.6a). That is, we may rewrite (6.8a) as

$$\widehat{z}^2(n) = \rho_1 z(n-1) + \rho_2\big(z(n-2) - \rho_1 z(n-1)\big) \tag{6.9a}$$

and, with a bit of algebra,

$$\rho_2 = \frac{p_2 - \rho_1 p_1}{(1-\rho_1^2)p_0}. \tag{6.9b}$$

Notice that ρ_2 can be obtained recursively from ρ_1 and values of P_E.

Analogous reasoning leads to the third-order LLS prediction:

$$\begin{aligned}\widehat{z}^3(n) = {} & \rho_1 z(n-1) + \rho_2\big(z(n-2) - \rho_1 z(n-1)\big) \\ & + \rho_3 \underbrace{\Big(z(n-3) - \rho_1 z(n-2) - \rho_2\big(z(n-2) - \rho_1 z(n-1)\big)\Big)}_{b_2}\end{aligned} \tag{6.10}$$

where b_2 is orthogonal to the first two right-hand side terms and the only quantity that needs to be computed is ρ_3. Just as with ρ_2, ρ_3 can be computed recursively and is based on the previously computed ρ_1 and ρ_2 and some elements of P_E. Continuing this order-recursive process, we ultimately arrive at the k-th order LLS prediction $\widehat{z}^k(n)$. Finally, the k-th order AR model is given by

$$z(n) = \widehat{z}^k(n) + \mu(n) \tag{6.11}$$

where the variance of the white-noise process $\mu(n)$ is

$$\mathrm{var}(\mu(n)) = p_0 \prod_{i=1}^{k} (1 - \rho_i^2) . \tag{6.12}$$

The parameters ρ_i are called *reflection coefficients* and are also known as *PARCOR coefficients* or *Schur coefficients* and have the following interpretation. The i-th reflection coefficient, ρ_i, is the correlation coefficient of $z(n)$ and $z(n+i)$ conditioned on the $i-1$ intervening values $\{z(j)\}_{j\in[n+1:n+i-1]}$. That is,

$$\rho_i = \frac{\mathrm{E}[\widetilde{z}(n)\widetilde{z}(n+i)]}{\mathrm{var}(\widetilde{z}(n))^{1/2}\,\mathrm{var}(\widetilde{z}(n+i))^{1/2}} \tag{6.13}$$

where

$$\widetilde{z}(n) \triangleq z(n) - \widehat{\mathrm{E}}[z(n) \mid \{z(j)\}_{j\in[n+1:n+i-1]}] , \tag{6.14a}$$

$$\widetilde{z}(n+i) \triangleq z(n+i) - \widehat{\mathrm{E}}[z(n+i) \mid \{z(j)\}_{j\in[n+1:n+i-1]}] . \tag{6.14b}$$

Due to our assumption of stationarity, the reflection coefficients do not depend on n.

Levinson's algorithm can be used to compute recursively the reflection coefficients which, as shown, can be used to form LLS predictors order-recursively. An important fact about Levinson's algorithm is that it establishes a one-to-one correspondence between the covariance elements $\{p_i\}$ and the reflection coefficients $\{\rho_i\}$. Any valid set of covariance elements corresponds to a set of reflection coefficients, each of which is bounded above in magnitude by unity. Therefore, *any* set of $\{\rho_j\}_{j>k}$ with $|\rho_j| < 1$ yields a valid AR model corresponding to a particular completion of P_E. The maximum-entropy completion corresponds to setting $\rho_j = 0$ for $j > k$. Levinson's algorithm can be generalized in a variety of ways for application to extension problems other than the banded one we have considered. For nonstationary banded problems [117, 118], the reflection coefficients depend on n and, so, are doubly indexed. The extra computations required lead to a complexity of $O(k^2N)$ for computing the k parameters required at each temporal index of a length N process. By allowing the autoregressive order k to vary with n as well, Levinson's algorithm can be applied to block-banded extension problems [58,59]. Other generalizations are suggested by the extension problems considered in [11,33,88,102,120]. Precise characterizations of these problems and the corresponding generalized-Levinson algorithm are most easily described in graph-theoretic terms, and we shall return to these topics in Section 6.3 and Section 6.4 after a review of the relevant graph theory.

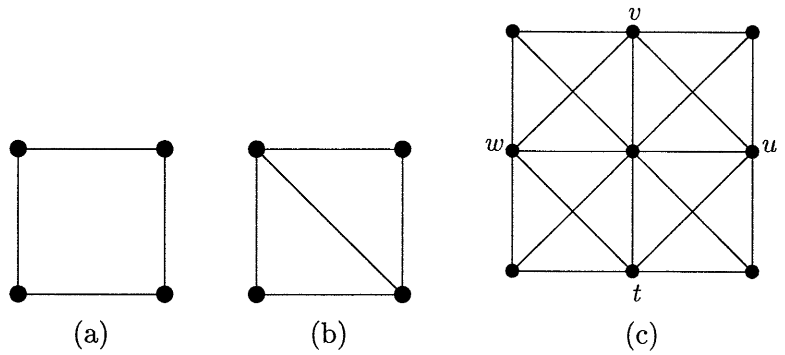


Figure 6.2. (a) Not chordal. (b) Chordal. (c) Not chordal (cycle $[t, u, v, w, t]$ has no chord).

■ 6.2 Some Graph Theory

In this section we introduce the aspects of graph theory required for the remainder of this chapter. For more on graph theory see [18,19,33,85,115,146,186]. A graph G is an ordered pair of sets $G = (V, E)$ where $E \subseteq V \times V$. V is the set of *vertices* of G and E is the set of *edges* of G. We will use the notation $\text{vert}(G) \triangleq V$ and $\text{edge}(G) \triangleq E$. There is no loss of generality in assuming that $V = \{0, 1, \ldots, |V| - 1\}$. An edge of a graph will be denoted by the pair (a, b) where a and b are vertices. All graphs considered in this chapter are *undirected* meaning that if (a, b) is an edge so is (b, a) so that E is a symmetric subset of $V \times V$. This is equivalent to assuming that (a, b) is an *unordered* pair, i.e., that $(a, b) = (b, a)$. A graph is *complete* if $E = V \times V$.

Associated with any graph $G = (V, E)$ is a unique $|V| \times |V|$ matrix called the *adjacency matrix.* The element in the i-th row and j-th column of the adjacency matrix for G is one if $(i, j) \in E$ and zero otherwise. There is a notion of equivalence among graphs which can be expressed in terms of adjacency matrices.

Definition 6.2.1 (Isomorphism). *Let G^i be a graph associated with adjacency matrix A^i for $i = 1, 2$. Then G^1 and G^2 are said to be* isomorphic *(to one another) if $A^1 = RA^2R^T$ for some permutation matrix R.*

For a graph $G = (V, E)$, a *path* of length n from v_0 to v_n is a sequence of distinct vertices $[v_0, v_1, \ldots, v_n]$ such that $(v_{i-1}, v_i) \in E$. For example, referring to the graph of Figure 6.2(c), $[t, u, v]$ is a path from vertex 1 to vertex 3. A graph is said to be *connected* if for every pair of vertices in the graph there is a path between them. All graphs of Figure 6.2 are connected.

A *cycle* of length $n + 1$ is a sequence of vertices $[v_0, v_1, \ldots, v_n, v_0]$ where the subsequence $[v_0, v_1, \ldots, v_n]$ is a path (of length n) and $(v_n, v_0) \in E$. For the graph of Figure 6.2(c), $[t, u, v, w, t]$ is a cycle. Throughout this chapter, unless indicated otherwise, we assume that a graph contains all cycles of length one (self-loops). That is, $(v, v) \in E$ for all $v \in V$. A cycle $[v_0, v_1, \ldots, v_n, v_0]$ is said to have a *chord* if $(v_i, v_j) \in E$ for $1 < |i - j| < n$. The cycle $[t, u, v, w, t]$ of the graph in Figure 6.2(c) has no chord.

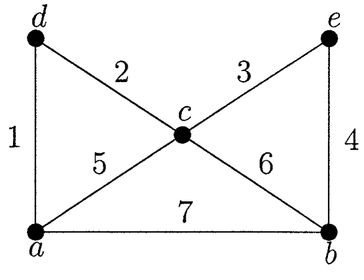


Figure 6.3. The numbers indicate the order in which edges are added to form a sequence of chordal graphs.

Adding either the edge (t, v) or (u, w) to the graph of Figure 6.2(c) would provide the cycle $[t, u, v, w, t]$ with a chord. There are two classes of graphs that are characterized by properties of their cycles and which are of particular relevance to this chapter: trees and chordal graphs.

Definition 6.2.2 (Tree). *A graph $G = (V, E)$ is called a* tree *if it is connected and has no cycles.*

It is well-known [146, 186] that $G = (V, E)$ is a tree if and only if $|V| = |E| + 1$.

Definition 6.2.3 (Chordal Graph). *A graph is called* chordal *(also called* triangulated*) if all cycles of length greater than three have a chord.*

Examples of chordal and non-chordal graphs are illustrated in Figure 6.2.

As will be explained in detail in Section 6.3, the set of edges, E, of a graph $G = (V, E)$ specifies the known elements of a partial covariance matrix. Covariance extension corresponds to adding edges. In this chapter we are interested in constructing extensions one element at a time, which corresponds to a sequence of graphs where each subsequent graph has a new edge. We shall see that sequences of chordal graphs have properties crucial to the extension problem.

Definition 6.2.4 ((Complete) Chordal Sequence). *Let $G^i = (V, E^i)$ be a chordal graph for $i \in \{0, 1, \ldots, n\}$. Then $[G^0, G^1, \ldots G^n]$ is a* chordal sequence *if $E^i = E^{i-1} \cup e^i$ with $e^i \notin E^{i-1}$. If, in addition, G^n is complete then $[G^0, G^1, \ldots G^n]$ is a* complete chordal sequence.

The following important proposition pertains to chordal sequences.

Proposition 6.2.1 ([33]). *If $G = (V, E)$ and $G' = (V, E')$ are chordal graphs with $E \subseteq E'$ then there exists a chordal sequence $[G, \ldots, G']$ between them.*

Figure 6.3 illustrates a chordal sequence. The edge labels indicate the sequence in which edges are added (i.e., edge labeled "1" is added first, edge labeled "2" is added second, etc.). We note that a chordal sequence $[G, \ldots, G']$ that begins with $G = (V, E)$ and ends with $G' = (V, E')$ is not unique. Proposition 6.2.1 does not address the problem of efficiently finding a chordal sequence between G and G'. However, for the special case in which G' is the complete graph, there exists an algorithm with complexity $O(|V|+|E|)$ for finding one [88, 158]. This algorithm is also easily extended to the special case for which G contains no edges other than self-loops. For more general circumstances, one may always find a chordal sequence $[G^0, G^1, \ldots, G^n]$ between chordal graphs G^0 and G^n by forming graph $G^i = (V, E^i)$ by searching over $E^n - E^{i-1}$ for an edge that preserves chordality. For each candidate edge in $E^n - E^{i-1}$, the complexity of checking chordality is no larger than $O(|V| + |E^i|)$ [177].

If $U \subseteq V$ then by $G_U = (U, E_U)$ we denote the subgraph of G induced by U where

$$E_U = \{(u, v) \in E \mid u, v \in U\} = E \cap (U \times U)\,. \tag{6.15}$$

A special type of subgraph is given in the following definition.

Definition 6.2.5 (Clique). *U is called a* clique *if G_U is a complete subgraph.*

Notice that, by definition, a clique is *not* a complete subgraph, rather it is a set of *vertices* that *induces* a complete subgraph. A clique is *maximal* if it is not a proper subset of another clique. Referring to the graph of Figure 6.3, the set $\{a, b\}$ is a clique but is not maximal. However, the set $\{a, b, c\}$ is a maximal clique. As we develop in the sequel, covariance extension will be associated with the maximal cliques of chordal graphs. One important result upon which we will rely is the following.

Lemma 6.2.1 ([88]). *Let $G^1 = (V, E^1)$ and $G^2 = (V, E^2)$ be chordal graphs. Suppose $E^2 = E^1 \cup e$ where $e = (a, b) \notin E^1$. Then the unique maximal clique of G^2 containing a and b is of the form $Q = \{a, b\} \cup (A \cap B)$ where A, B are maximal cliques of G^1, such that $a \in A$ and $b \in B$.*

In the preceding lemma, we have created a new chordal graph G^2 with a new maximal clique Q out of a chordal graph G^1 by adding a new edge $e = (a, b)$. We will call the end-points a and b of e the *active* elements of the maximal clique Q. This is illustrated with a specific example in Figure 6.3. Let $G^1 = (V, E^1)$ be the chordal graph for which $V = \{a, b, c, d, e\}$ and where E^1 includes edges 1–6 as depicted in Figure 6.3. Let $G^2 = (V, E^2)$ where E^2 contains edges 1–7. When the seventh edge (a, b) is added to form graph G^2, the maximal clique containing the active elements a and b is $Q = \{a, b, c\}$. Notice that Q can be written as $Q = \{a, b\} \cup (A \cap B)$ where $A = \{a, c, d\}$ and $B = \{b, c, e\}$, both maximal cliques of the G^1.

Consider the set $\mathcal{K}$ of maximal cliques of some graph G and let T be a tree whose vertex set is $\mathcal{K}$. Such a tree is called a *clique tree* or *junction tree* if it has the following intersection property.

Definition 6.2.6 (Intersection Property). *A tree $T = (\mathcal{K}, \mathcal{E})$ with $\mathcal{K} = \{F_i\}$, a family of sets, has the* intersection property *if $F_i \cap F_j \subseteq F_k$ whenever F_k lies on the (unique) path from F_i to F_j.*

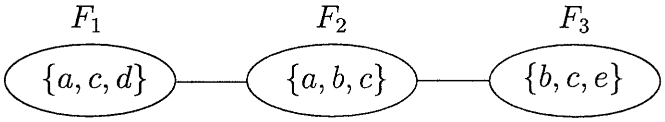


Figure 6.4. A junction tree for the graph of Figure 6.3.

Definition 6.2.7 (Junction Tree). *Let $G = (V, E)$ be a graph and let $\mathcal{K}$ be the set of maximal cliques of G. A tree $T = (\mathcal{K}, \mathcal{E})$ whose vertices are the maximal cliques of G is called a* junction tree *if it has the intersection property.*

A junction tree for the graph of Figure 6.3 is illustrated in Figure 6.4. Notice that the vertex labeled $F_2 = \{a,b,c\}$ is on the path between $F_1 = \{a,c,d\}$ and $F_3 = \{b,c,e\}$ and contains their intersection. It is shown in [33] that an equivalent definition of a junction tree is given as follows.

Definition 6.2.8 (Junction Tree II). *A* junction tree *for a graph $G = (V, E)$ is a tree $T = (\mathcal{K}, \mathcal{E})$ whose vertex set $\mathcal{K}$ is the set of maximal cliques of G and where for any $v \in V$, each induced subgraph $T_{\mathcal{K}_v}$ is connected (and, hence, a subtree), where $\mathcal{K}_v$ consists of those maximal cliques of G that contain v.*

The following provides a connection between chordal graphs and junction trees.

Proposition 6.2.2 ([85]). *G is chordal if and only if there exists a junction tree for G.*

For *any* graph $J = (\mathcal{L}, \mathcal{E})$ such that $\mathcal{L}$ is a family of sets $\mathcal{L} = \{F_i\}_{i=1}^{\ell}$, we will denote by ucli$(J)$ the set

$$\text{ucli}(J) = \bigcup_{i=1}^{\ell} F_i . \tag{6.16}$$

Notice that when J is a junction tree for some graph $G = (V, E)$ then ucli(J) is equal to V. The notation ucli$(\cdot)$ is intended to remind us of this fact and stands for "union of cliques."

■ 6.3 Covariance Extension and Chordal Graphs

In Section 6.1 we considered the partial covariance matrix P_E where $E = \{(m,n) \mid |m-n| \leq k\}$. In this section we consider more general support sets. Any support set $E^0 \subset V \times V$ can be associated with an undirected graph $G^0 = (V, E^0)$ that contains all self-loops (which are not included in our figures). In this way, covariance extensions can be conveniently described graphically as follows. Extending P_{E^0} by defining one new covariance element with indices $(a,b) \notin E^0$ corresponds to adding the new edge $e^1 \triangleq (a,b)$ to G^0. If we define $G^1 = (V, E^1)$ where $E^1 = E^0 \cup e^1$ then, in making this one-element extension, we have obtained a new partial covariance matrix P_{E^1}. More generally, an extension from P_{E^0} to P_{E^n} corresponds to adding n new covariance elements with indices $e^i = (a^i, b^i) \notin E^0$. These new indices represent edges $\{e^i\}_{i=1}^n$ of graph $G^n = (V, E^n)$ where $E^n = E^0 \bigcup_{i=1}^n e^i$. Conditions for the existence of an extension P_{E^n} of P_{E^0} are found in the literature and reviewed in this section.

We begin with the case for which $E^n = V \times V$ which corresponds to the problem of finding a completely specified covariance matrix P such that $P(i,j) = P_{E^0}(i,j)$ for all $(i,j) \in E^0$.

Proposition 6.3.1 ([33, 88]). *Completions exist for all values $\{P_{E^0}(i,j)\}_{(i,j)\in E^0}$ of the partial covariance matrix P_{E^0} if and only if $G^0 = (V, E^0)$ is a chordal graph.*

It is worth emphasizing that when G^0 is not chordal, there may exist completions for partial covariance matrices P_{E^0} for specific choices of the entries $\{P_{E^0}(i,j)\}_{(i,j)\in E^0}$. However, if the entries are unconstrained (other than what is required to satisfy $P_{E^0} > 0$), then Proposition 6.3.1 tells us that completions exist for *all* choices of the entries exactly when G^0 is chordal. Since, as shown in Section 6.1, completions exist for banded partial covariance matrices (cf., (6.4) and Figure 6.1), Proposition 6.3.1 shows that the graph $G = (V, E)$ where E is as given in (6.4) is chordal.

Consider now the case for which for which $E^0 \subset E^n \subset V \times V$. Extensions, P_{E^n}, of P_{E^0} exist when G^0 is chordal because we may simply restrict any completion (whose existence is guaranteed by Proposition 6.3.1) to E^n. This is not very satisfying, however, because it implies that one must first compute a completion (which requires inferring *all* of the unspecified covariance entries) even if one is interested in an extension to just a few elements. However, as we show shortly, under certain conditions there is a way to compute just the covariance entries of interest without computing all of them. Moreover, there is a way to compute the entries sequentially rather than in batch. In developing this sequential extension technique, we will rely on the following proposition.

Proposition 6.3.2 ([11]). *Let $G^0 = (V, E^0)$, $G^1 = (V, E^1)$ be chordal graphs and $e = (a, b) \notin E^0$ be an edge such that $E^1 = E^0 \cup e$. Let Q be the unique maximal clique of G^1 containing a, b. Let $G_Q^i = (Q, E_Q^i)$ be the sub-graph of G^i induced by Q for $i = 0, 1$ (see (6.15)). If P is the unique maximum-entropy completion of P_{E^0} and $P_{E_Q^1}$ is the unique maximum-entropy completion of $P_{E_Q^0}$. Then $P_{E_Q^1}(a, b) = P(a, b)$.*

Proposition 6.3.2 tells us that to find the maximum-entropy value of a specific unknown covariance element indexed by $e = (a, b)$, one need not find a full completion. Rather, it is sufficient to consider the completion of the maximal principal minor defined by the maximal clique Q of Proposition 6.3.2. We emphasize that each maximal clique of $G = (V, E)$ defines a maximal principal minor of P_E. The existence and uniqueness of a maximal clique Q as defined in Proposition 6.3.2 is guaranteed by Lemma 6.2.1. Since, in Proposition 6.3.2, G^0 is chordal, so is G_Q^0, hence, completions of P_{E^0} and $P_{E_Q^0}$ exist by Proposition 6.3.1. The existence and uniqueness of maximum-entropy completions P and $P_{E_Q^1}$ is guaranteed by the following proposition.

Proposition 6.3.3 ([88]). *If any completions of P_{E^0} exist then there exists a unique completion with maximal entropy.*

Proposition 6.3.2 suggests that we can build up a completion one element at a time by considering a complete chordal sequence $[G^0, G^1, \ldots, G^n]$. Associated with this sequence is a sequence of edges $\{e^i\}_{i=1}^n$ where $e^i = (a^i, b^i)$ is added to G^{i-1} to form G^i. This sequence of edges, in turn, defines a sequence of new maximal cliques $\{Q^i\}$ where a^i, b^i are the active elements of Q^i. By Proposition 6.3.2, we may find the (a^i, b^i) element of the maximum-entropy completion of $P_{E^{i-1}}$ by restricting the problem to the vertices in Q^i and solving this smaller completion problem. The same idea is also discussed in [33, 88].

Building up the maximum-entropy completion with this sequence of one-element extensions turns a seemingly highly nonlinear problem into a sequence of quadratic maximizations each of which requires the solution of a linear equation. That is, at each step we maximize the determinant[7] of a matrix with just one unknown element indexed by $e^i = (a^i, b^i)$. Of course, we are free to terminate this process at any step and need not find every unknown element. An important property of this sequence of one-element extensions based on a chordal sequence is that each new covariance element depends only on ones that have been previously computed (or have been provided in P_{E^0}). Therefore, we may obtain extensions without computing the full completion as the following corollary states.

Corollary 6.3.1. *Let $G^0 = (V, E^0)$ and $G^n = (V, E^n)$ be chordal graphs with $E^0 \subset E^n$. Let P be the maximum-entropy extension of P_{E^0} then the elements of $\{P(i,j)\}_{(i,j)\in E^n}$ are not a function of $\{P(i,j)\}_{(i,j)\notin E^n}$.*

Proof. By Proposition 6.2.1 a complete chordal sequence exists that begins with the chordal sequence $[G^0, \ldots, G^n]$. Now iteratively apply Proposition 6.3.2 and terminate the iteration at step n. ■

While Corollary 6.3.1 pertains to maximum-entropy extensions, it can be generalized to all possible extensions. That is, if G^n and G^0 are as in Corollary 6.3.1 then any extension of P_{E^n} of P_{E^0} can be obtained via a sequence of one-element extensions associated with a chordal sequence where, at each step, the new covariance value is constrained only by the previously computed (or given) ones. A constructive proof of this fact is provided in the following section in which we present a generalization of the Levinson algorithm discussed in Section 6.1.

■ 6.4 A Generalized-Levinson Algorithm

In this section we provide a generalized-Levinson algorithm and illustrate how it can be applied to the problem of covariance completion and extension. The generalized-Levinson algorithm is applicable to a substantially broader class of completion problems

[7] As was stated in Section 6.1, maximizing entropy and maximizing the determinant are equivalent [88, 120].

than the classical Levinson algorithm. Indeed, it may be used to compute a completion of a given partial covariance matrix P_{E^0} corresponding to *any* chordal graph $G = (V, E^0)$. The completion is performed one element at a time, ensuring at each step that the obtained extension is positive-definite. This corresponds to a sequence of graphs $[G^0, G^1, \ldots]$ where $G^i = (V, E^i)$ has one more edge than G^{i-1}. It has been shown previously [33] that *any* completion may be obtained by such a sequence of one-element extensions if the corresponding sequence of graphs is a complete chordal sequence. Of course, terminating the complete chordal sequence prior to arriving at the complete graph results in an extension rather than a completion. Therefore, extensions P_{E^n} may be obtained for any chordal graph $G^n = (V, E^n)$ such that $E^0 \subset E^n$. The contribution of this section is to provide a recursive procedure for covariance completion.

We will show that the condition of positive-definiteness, i.e., $P_{E^i} > 0$, holds for all i if and only if each new covariance element added at each step i is in a certain range of values. At each step i, this range is parameterized by a (generalized) reflection coefficient which is completely determined by previously computed (or given) covariance elements $P_{E^{i-1}}$. That is, there is a one-to-one correspondence between the covariance elements and the reflection coefficients. The magnitude of each reflection coefficient is bounded above by unity and setting the reflection coefficients to zero leads to the maximum-entropy extension [33].

Since our approach to covariance completion is via a sequence of one-element extensions we need only describe an arbitrary step. To this end, suppose that P_E is a partial covariance matrix and $G = (V, E)$ is chordal. Consider computing the (currently undefined) covariance element indexed by $e = (a, b)$ where $a, b \in V$ and where the graph $G' = (V, F = E \cup e)$ is also chordal. That is, consider the one-element extension of P_E to P_F. Let $Q = \{a, b\} \cup (A \cap B)$ be the unique maximal clique of G' for which a, b are active where $a \in A$, $b \in B$ and A and B are maximal cliques of G (cf., Lemma 6.2.1). To simplify the notation of our subsequent development, we make the following definitions

$$U \triangleq A \cap B \,, \tag{6.17a}$$

$$U_a \triangleq \{a\} \cup U = \{a\} \cup (A \cap B) \,, \tag{6.17b}$$

$$U_b \triangleq \{b\} \cup U = \{b\} \cup (A \cap B) \,. \tag{6.17c}$$

So, using these definitions, $Q = \{a, b\} \cup U = U_a \cup U_b$ and $U = U_a \cap U_b$.

In the following example we introduce a simple covariance extension problem to which we will return in our subsequent discussion. The example also illustrates the relationships among the sets Q, U, U_a, and U_b.

Example: 4×4 Banded Partial Covariance Matrix

Consider the 4×4 partial covariance matrix whose rows and columns are indexed by $\{a, u, b, v\}$ as illustrated in Figure 6.5. The filled circles indicate elements that are known and the question marks indicate unknown elements. In the sequel we will use this simple case to illustrate the application of our generalized-Levinson algorithm to

	a	u	b	v
a	●	●	?	?
u	●	●	●	?
b	?	●	●	●
v	?	?	●	●

Figure 6.5. 4 × 4 banded partial covariance matrix. Filled circles represent known elements, question marks represent unknown ones.

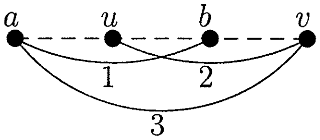


Figure 6.6. Chordal sequence for completing the matrix of Figure 6.5. The edge sequence is indicated by the numbered labels: graph G^1 is formed by the addition of edge "1", graph G^2 by the addition of edge "2," and graph G^3 by the addition of edge "3."

covariance completion. The sequence of chordal graphs which will be used to accomplish the covariance completion is illustrated in Figure 6.6. The dashed lines indicate the original graph, G^0, and the solid lines indicate the sequence of additional edges. The edge labeled "1" is added first to form graph G^1, the edge labeled "2" is added second to form graph G^2, and the edge labeled "3" is added last to form graph G^3.

Consider computing the element indexed by (a, b). Computing this element corresponds to adding the edge labeled "1" in Figure 6.6 to form graph G^1. The new maximal clique of G^1 formed for which a and b are active elements is $Q = \{a, u, b\}$. Notice that Q can also be written as $Q = \{a, b\} \cup (\{a, u\} \cap \{u, b\})$ where the sets $\{a, u\}$ and $\{u, b\}$ are maximal cliques of the original graph, G^0. Hence, for this one-element extension step, $\{a, u\}$ plays the role of A which, in this case, is equal to U_a, while $\{u, b\}$ plays the role of B which, in this case, is equal to U_b, and $\{u\}$ plays the role of $U = U_a \cap U_b$.

■ 6.4.1 Generalized Reflection Coefficients

Our first objective is to characterize the range of possible values of the new covariance element, which we will denote by $p_{a,b}$, so that P_F is positive-definite. Because maximal principal minors correspond to maximal cliques,[8] $P_F > 0$ if (and only if) the maximal principal minor given by

$$P_{Q^2} = \{P_F(i,j) \mid (i,j) \in Q \times Q = Q^2\} \tag{6.18}$$

[8] Recall that by $P_F > 0$ we mean that all maximal principal minors are positive-definite.

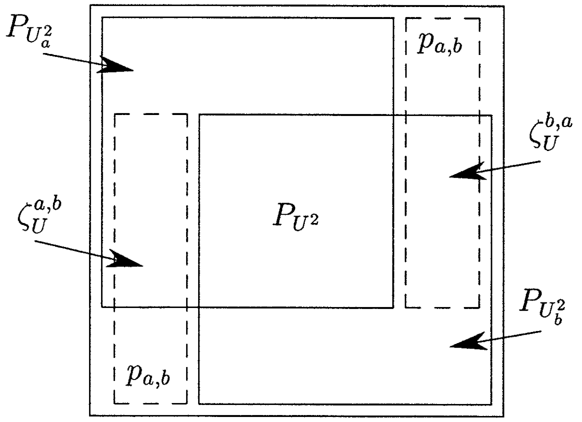


Figure 6.7. The maximal principal minor P_{Q^2} contains principal minors $P_{U_a^2}$ (upper left), $P_{U_b^2}$ (lower right), and P_{U^2} (center). The vectors $\zeta_U^{a,b}$ and $\zeta_U^{b,a}$ and the element $p_{a,b}$ are also indicated.

is positive-definite. Indeed, it is assumed that $P_E > 0$, and since Q is the unique maximal clique containing the new edge e, it is sufficient and necessary that $P_{Q^2} > 0$. Since Q contains the sets U, U_a and U_b, the maximal principal minor P_{Q^2} contains the principal minors P_{U^2}, $P_{U_a^2}$ and $P_{U_b^2}$ where these are defined analogously to P_{Q^2}. That is,

$$P_{U^2} = \{P_F(i,j) \mid (i,j) \in U \times U = U^2\}, \tag{6.19a}$$

$$P_{U_a^2} = \{P_F(i,j) \mid (i,j) \in U_a \times U_a = U_a^2\}, \tag{6.19b}$$

$$P_{U_b^2} = \{P_F(i,j) \mid (i,j) \in U_b \times U_b = U_b^2\}. \tag{6.19c}$$

These principal minors, as well as some of the other notation to be used in this section, are illustrated in Figure 6.7. The matrix $P_{U_a^2}$ contains all but last row and column of P_{Q^2} and is indicated in upper left of Figure 6.7; $P_{U_b^2}$ contains all but first row and column of P_{Q^2} and is indicated in lower right of Figure 6.7; and P_{U^2} contains all but the first and last row and column of P_{Q^2}, is the intersection of $P_{U_a^2}$ and $P_{U_b^2}$, and is indicated in the center of Figure 6.7. The only unknown covariance element $p_{a,b}$ occupies the upper right and lower left corners of P_{Q^2}.

After introducing some basic concepts and notation which we will use in both this section and the next, we will provide several propositions which characterize the range of values of $p_{a,b}$ and the maximum-entropy value. To this end, let z be a zero-mean random vector indexed by Q with covariance matrix P_{Q^2}. We will denote by z_D the sub-vector of z indexed by $D \subset Q$. When $D = \{d\}$ is a singleton set, we will often write z_d rather than $z_{\{d\}}$. Consider the LLS estimate of the scalar random variable z_b based on the random vector z_U which has the form

$$\hat{z}_b = L_U^b z_U \,. \tag{6.20}$$

The row-vector L_U^b satisfies the so-called *normal equations* which are derived as follows. Because $\widehat{z}_b - z_b$ is orthogonal to z_U we have

$$\begin{aligned}
\begin{bmatrix}0 & \cdots & 0\end{bmatrix} &= \mathrm{E}[(z_b - \widehat{z}_b) z_U^T] && (6.21\text{a})\\
&= \mathrm{E}[(z_b - L_U^b z_U) z_U^T] && (6.21\text{b})\\
&= \begin{bmatrix}-L_U^b & 1\end{bmatrix} \mathrm{E}\left\{ \begin{bmatrix} z_U \\ z_b \end{bmatrix} z_U^T \right\}. && (6.21\text{c})
\end{aligned}$$

By defining ε_U^b to be the LLS estimation error variance we have

$$\begin{aligned}
\varepsilon_U^b &= \mathrm{E}[(z_b - \widehat{z}_b)^2] && (6.22\text{a})\\
&= \mathrm{E}[(z_b - \widehat{z}_b) z_b] && (6.22\text{b})\\
&= \mathrm{E}[(z_b - L_U^b z_U) z_b] && (6.22\text{c})\\
&= \begin{bmatrix}-L_U^b & 1\end{bmatrix} \mathrm{E}\left\{ \begin{bmatrix} z_U \\ z_b \end{bmatrix} z_b \right\}. && (6.22\text{d})
\end{aligned}$$

Combining (6.21) and (6.22) we obtain the normal equations:

$$\begin{aligned}
\begin{bmatrix}0 & \cdots & 0 & \varepsilon_U^b\end{bmatrix} &= \begin{bmatrix}-L_U^b & 1\end{bmatrix} \mathrm{E}\left\{ \begin{bmatrix} z_U \\ z_b \end{bmatrix} \begin{bmatrix} z_U^T & z_b \end{bmatrix} \right\} && (6.23\text{a})\\
&= \underbrace{\begin{bmatrix}-L_U^b & 1\end{bmatrix}}_{B_U^b} P_{U_b^2} && (6.23\text{b})
\end{aligned}$$

where B_U^b is as defined in (6.23). By augmenting B_U^b with a zero and expanding $P_{U_b^2}$ by one row and column we have

$$\begin{bmatrix}0 & B_U^b\end{bmatrix} P_{Q^2} = \begin{bmatrix}\delta_U^{a,b} & 0 & \cdots & 0 & \varepsilon_U^b\end{bmatrix} \tag{6.24a}$$

where

$$\delta_U^{a,b} \triangleq B_U^b \zeta_U^{a,b} \tag{6.24b}$$

and

$$\zeta_U^{a,b} \triangleq \mathrm{E}\left\{ \begin{bmatrix} z_U \\ z_b \end{bmatrix} z_a \right\} = \begin{bmatrix} \mathrm{E}[z_U z_a] \\ p_{a,b} \end{bmatrix}. \tag{6.24c}$$

Notice that only the last element of $\zeta_U^{a,b}$ depends on $p_{a,b}$ and the remaining elements belong to the previously defined principal minor $P_{U_b^2}$ (see Figure 6.7). We can, therefore, relate $\delta_U^{a,b}$ to $p_{a,b}$ as follows. Using (6.24c) and the definition of B_U^b given in (6.23),

$$\delta_U^{a,b} = -L_U^b\, \mathrm{E}[z_U z_a] + p_{a,b}\,. \tag{6.25}$$

In the sequel we will provide a bound on $\delta_U^{a,b}$ which, in turn, will bound the range of values for $p_{a,b}$.

Next, we consider the LLS estimate of the scalar random variable z_a based on the random vector z_U which has the form

$$\widehat{z}_a = L_U^a z_U \,. \tag{6.26}$$

Using reasoning similar to that of the previous paragraph, the row-vector L_U^a must satisfy the following normal equations:

$$\underbrace{\begin{bmatrix}1 & -L_U^a\end{bmatrix}}_{A_U^a} P_{U_a^2} = \begin{bmatrix}\varepsilon_U^a & 0 & \cdots & 0\end{bmatrix} \tag{6.27}$$

where ε_U^a is the estimation error variance, and A_U^a is as defined in (6.27). By augmenting A_U^a with a zero element and expanding $P_{U_a^2}$ by one row and column, we have

$$\begin{bmatrix}A_U^a & 0\end{bmatrix} P_{Q^2} = \begin{bmatrix}\varepsilon_U^a & 0 & \cdots & 0 & \delta_U^{b,a}\end{bmatrix} \,. \tag{6.28a}$$

where

$$\delta_U^{b,a} \triangleq A_U^a \zeta_U^{b,a} \tag{6.28b}$$

and

$$\zeta_U^{b,a} \triangleq \mathrm{E}\left\{\begin{bmatrix}z_a \\ z_U\end{bmatrix} z_b\right\} = \begin{bmatrix}p_{a,b} \\ \mathrm{E}[z_U z_b]\end{bmatrix} \,. \tag{6.28c}$$

The quantity $\delta_U^{b,a}$ is related to $p_{a,b}$ by

$$\delta_U^{b,a} = -L_U^a \, \mathrm{E}[z_U z_b] + p_{a,b} \tag{6.29}$$

where we have used (6.28c) and the definition of A_U^a given in (6.27). The following proposition shows the equivalence of (6.25) and (6.29).

Lemma 6.4.1. *$\delta_U^{a,b}$ and $\delta_U^{b,a}$, of* (6.25) *and* (6.29)*, respectively, are equal.*

Proof. Using (6.28a) and the fact that $\begin{bmatrix}0 & B_U^b\end{bmatrix} = \begin{bmatrix}0 & -L_U^b & 1\end{bmatrix}$ we have

$$\begin{bmatrix}A_U^a & 0\end{bmatrix} P_{Q^2} \begin{bmatrix}0 & B_U^b\end{bmatrix}^T = \delta_U^{b,a} \,. \tag{6.30}$$

On the other hand, using (6.24a) and the fact that $\begin{bmatrix}A_U^a & 0\end{bmatrix} = \begin{bmatrix}1 & -L_U^a & 0\end{bmatrix}$, we have

$$\begin{bmatrix}0 & B_U^b\end{bmatrix} P_{Q^2} \begin{bmatrix}A_U^a & 0\end{bmatrix}^T = \delta_U^{a,b} \,. \tag{6.31}$$

From (6.30) and (6.31) it follows that $\delta_U^{a,b} = \delta_U^{b,a}$. ■

The following corollary follows immediately.

Corollary 6.4.1. *The quantities* $L_U^b \operatorname{E}[z_U z_a]$ *and* $L_U^a \operatorname{E}[z_U z_b]$ *which appear in* (6.25) *and* (6.29), *respectively, are equal.*

To characterize the range of possible values of $p_{a,b}$ it will be convenient to define the (generalized) reflection coefficient, $\rho_U^{a,b}$, as

$$\rho_U^{a,b} \triangleq \frac{\delta_U^{a,b}}{\sqrt{\varepsilon_U^a \varepsilon_U^b}}. \tag{6.32}$$

We can rewrite (6.25) and (6.29) in terms of $\rho_U^{a,b}$ and, with a bit of rearrangement, obtain

$$\begin{aligned} p_{a,b} &= L_U^b \operatorname{E}[z_U z_a] + \rho_U^{a,b} \sqrt{\varepsilon_U^a \varepsilon_U^b} && \text{(6.33a)} \\ &= L_U^a \operatorname{E}[z_U z_b] + \rho_U^{a,b} \sqrt{\varepsilon_U^a \varepsilon_U^b}. && \text{(6.33b)} \end{aligned}$$

All quantities on the right-hand side of (6.33a) and in (6.33b) other than $\rho_U^{a,b}$ can be deduced from the values of P_{Q^2} that are known. That is, they can be determined without knowledge of $p_{a,b}$. Therefore, through (6.33), $\rho_U^{a,b}$ parameterizes all possible values of $p_{a,b}$. Next, we provide a bound on the magnitude of $\rho_U^{a,b}$ that, in turn, characterizes all possible values of $p_{a,b}$ that are consistent with the known values of P_{Q^2}. The following proposition, together with (6.33), establishes a one-to-one correspondence between $\rho_U^{a,b}$ and any valid $p_{a,b}$.

Proposition 6.4.1. *Using the notation defined previously,* $P_{Q^2} > 0$ *if and only if* $|\rho_U^{a,b}| < 1$. *The choice of* $\rho_U^{a,b} = 0$ *maximizes the determinant of* P_{Q^2}.

Proof. To show that $|\rho_U^{a,b}|$ cannot exceed unity when $P_{Q^2} > 0$, it suffices to show that $\rho_U^{a,b}$ is a correlation coefficient. Therefore, by (6.32), we need to show that $\delta_U^{a,b} = \operatorname{E}[(z_a - \widehat{z}_a)(z_b - \widehat{z}_b)]$. This is easily done as follows:

$$\begin{aligned} \operatorname{E}[(z_a - \widehat{z}_a)(z_b - \widehat{z}_b)] &= \operatorname{E}[(z_a - \widehat{z}_a)z_b] && \text{(6.34a)} \\ &= p_{a,b} - L_U^a \operatorname{E}[z_U z_b] && \text{(6.34b)} \\ &= \delta_U^{a,b} && \text{(6.34c)} \end{aligned}$$

where in (6.34a) we have used the fact that $z_a - \widehat{z}_a$ is orthogonal to z_U and, hence, to $\widehat{z}_b$; in (6.34b) we have taken expectations; in (6.34c) we have used (6.29) and Lemma 6.4.1. A different proof of the fact that $|\rho_U^{a,b}|$ is bounded by unity and the completion of the proof of Proposition 6.4.1 is found in Appendix D. ∎

There are two important implications of Proposition 6.4.1. First, the range of values for $\rho_U^{a,b}$, namely $(-1,1)$, provides a range of possible values for $p_{a,b}$ through (6.33). Second, the choice of $\rho_U^{a,b} = 0$ provides the determinant-maximizing value of $p_{a,b}$. Since, as stated previously, determinant maximization and entropy maximization coincide,

$$p_{a,b} = p_{a,b}^{\mathrm{ME}} \triangleq L_U^b \, \mathrm{E}[z_U z_a] \,. \tag{6.35}$$

is the entropy-maximizing value of $p_{a,b}$. Using Corollary 6.4.1 we can write this equivalently as

$$p_{a,b}^{\mathrm{ME}} \triangleq L_U^a \, \mathrm{E}[z_U z_b] \,. \tag{6.36}$$

Our final expression for $p_{a,b}$ is

$$p_{a,b} = p_{a,b}^{\mathrm{ME}} + \rho_U^{a,b} \sqrt{\varepsilon_U^a \varepsilon_U^b} \,. \tag{6.37}$$

We now continue with the example begun on page 123 and illustrate the one-element extension step.

Example: 4×4 Banded Partial Covariance Matrix, Continued

Consider again the partial covariance matrix illustrated in Figure 6.5 and the sequence of chordal graphs used to complete it as illustrated in Figure 6.6. We first discuss determining a value for the element indexed by (a,b) which we denote by $p_{a,b}$. This corresponds to adding the edge labeled "1" in Figure 6.6 to form graph G^1. Recall that the new maximal clique of G^1 for which a and b are active elements is $Q = \{a,u,b\} = \{a,b\} \cup (\{a,u\} \cap \{u,b\})$ so that $\{u\}$ plays the role of U. Thus, all possible values of $p_{a,b}$ are given by (6.33) in which the U is replaced by the set $\{u\}$. That is, all possible values of $p_{a,b}$ are given by

$$p_{a,b} = L_{\{u\}}^b \, \mathrm{E}[z_{\{u\}} z_a] + \rho_{\{u\}}^{a,b} \sqrt{\varepsilon_{\{u\}}^a \varepsilon_{\{u\}}^b} \tag{6.38a}$$

$$= L_{\{u\}}^a \, \mathrm{E}[z_{\{u\}} z_b] + \rho_{\{u\}}^{a,b} \sqrt{\varepsilon_{\{u\}}^a \varepsilon_{\{u\}}^b} \,. \tag{6.38b}$$

The quantities $A_{\{u\}}^a$ (equivalently, $L_{\{u\}}^a$), $B_{\{u\}}^b$ (equivalently, $L_{\{u\}}^b$), $\varepsilon_{\{u\}}^a$, $\varepsilon_{\{u\}}^b$ are functions of the given covariance data and are readily computed. By choosing a value for $\rho_{\{u\}}^{a,b}$ we fix $p_{a,b}$. A similar procedure is applied to determine a value for $p_{u,v}$ and corresponds to adding the edge labeled "2" in Figure 6.6 to form graph G^2. The new maximal clique formed in this step $\{u,b,v\} = \{u,v\} \cup (\{u,b\} \cap \{b,v\})$ where $\{u,b\}$ and $\{b,v\}$ are maximal cliques of G^1. Therefore, to determine $p_{u,v}$ we again use (6.33) with a replaced by u, with b replaced by v, and with U replaced by $\{b\}$. To do so, the quantities $A_{\{b\}}^u$ (equivalently, $L_{\{b\}}^u$), $B_{\{b\}}^v$ (equivalently, $L_{\{b\}}^v$), $\varepsilon_{\{b\}}^u$, $\varepsilon_{\{b\}}^v$ are first computed from the original given covariance data and the reflection coefficient $\rho_{\{b\}}^{u,v}$ is selected.

■ 6.4.2 Generalized-Levinson Recursion

The procedure described in the previous section is one step in a sequence of one-element extensions corresponding to a sequence of chordal graphs. In performing the one-element extension corresponding to (a,b), the quantities A_U^a, B_U^b, ε_U^a, and ε_U^b are computed and $\rho_U^{a,b}$ is selected. Having determined these quantities, they may then be used to solve higher-order systems of (normal) equations[9] which may arise in subsequent one-element extension steps. This leads to the *generalized-Levinson recursion* which we describe in this section. We emphasize that the quantities A_U^a, B_U^b, ε_U^a, ε_U^b, and $\rho_U^{a,b}$ have been computed previously. In particular, the first four quantities have been constructed based on elements of the original partial covariance matrix, P_{E^0}, and elements computed in previous steps in the recursion. The quantity $\rho_U^{a,b}$ is selected at will. The quantities A_U^a, B_U^b, ε_U^a, ε_U^b appear in the normal equations (6.23) and (6.27) which we repeat here for the reader's convenience:

$$\underbrace{\begin{bmatrix}-L_U^b & 1\end{bmatrix}}_{B_U^b} P_{U_b^2} = \begin{bmatrix}0 & \cdots & 0 & \varepsilon_U^b\end{bmatrix}\,, \tag{6.39a}$$

$$\underbrace{\begin{bmatrix}1 & -L_U^a\end{bmatrix}}_{A_U^a} P_{U_a^2} = \begin{bmatrix}\varepsilon_U^a & 0 & \cdots & 0\end{bmatrix}\,. \tag{6.39b}$$

Recall the augmented normal equations (6.24a) and (6.28a) which we repeat:

$$\begin{bmatrix}0 & B_U^b\end{bmatrix} P_{Q^2} = \begin{bmatrix}\delta_U^{a,b} & 0 & \cdots & 0 & \varepsilon_U^b\end{bmatrix}\,, \tag{6.40a}$$

$$\begin{bmatrix}A_U^a & 0\end{bmatrix} P_{Q^2} = \begin{bmatrix}\varepsilon_U^a & 0 & \cdots & 0 & \delta_U^{b,a}\end{bmatrix} \tag{6.40b}$$

where, as we have shown, $\delta_U^{a,b} = \delta_U^{b,a}$.

Equation (6.39a) and (6.40a) are concerned with the LLS estimate of z_b given z_U which arises in the one-element extension step involving element $p_{a,b}$. We now consider the higher order normal equations associated with the LLS estimate of z_b given $z_{U_a} = \begin{bmatrix}z_a & z_U^T\end{bmatrix}^T$. The estimator has the form $L_{U_a}^b z_{U_a}$ where the row-vector $L_{U_a}^b$ satisfies the normal equations

$$\underbrace{\begin{bmatrix}-L_{U_a}^b & 1\end{bmatrix}}_{B_{U_a}^b} P_{Q^2} = \begin{bmatrix}0 & \cdots & 0 & \varepsilon_{U_a}^b\end{bmatrix} \tag{6.41}$$

where $\varepsilon_{U_a}^b$ is the estimation error variance and $B_{U_a}^b$ is as defined in (6.41). The following proposition shows that $\varepsilon_{U_a}^b$ and $L_{U_a}^b$ (or, equivalently, $B_{U_a}^b$) are uniquely determined by the previously computed quantities B_U^b, A_U^b, ε_U^a, ε_U^b, and $\rho_U^{a,b}$.

[9] By a higher-order system of equations we mean one involving more variables.

Proposition 6.4.2. *Using the notation previously defined,*

$$\underbrace{\left[-L_{U_a}^b \quad 1\right]}_{B_{U_a}^b} = \left[0 \quad B_U^b\right] - \rho_U^{a,b}\sqrt{\frac{\varepsilon_U^b}{\varepsilon_U^a}}\left[A_U^a \quad 0\right] \tag{6.42a}$$

and

$$\varepsilon_{U_a}^b = \varepsilon_U^b\left(1 - \left(\rho_U^{a,b}\right)^2\right). \tag{6.42b}$$

Proof. Using the augmented normal equations (6.40) we have

$$\left\{\left[0 \quad B_U^b\right] - \frac{\delta_U^{a,b}}{\varepsilon_U^a}\left[A_U^a \quad 0\right]\right\}P_{Q^2} = \left[0 \quad \cdots \quad 0 \quad \varepsilon_U^b - \frac{\left(\delta_U^{a,b}\right)^2}{\varepsilon_U^a}\right] \tag{6.43a}$$

$$= \left[0 \quad \cdots \quad 0 \quad \varepsilon_U^b\left(1 - \left(\rho_U^{a,b}\right)^2\right)\right]. \tag{6.43b}$$

Hence, we conclude that

$$\underbrace{\left[-L_{U_a}^b \quad 1\right]}_{B_{U_a}^b} = \left[0 \quad B_U^b\right] - \frac{\delta_U^{a,b}}{\varepsilon_U^a}\left[A_U^a \quad 0\right] \tag{6.44a}$$

$$= \left[0 \quad B_U^b\right] - \rho_U^{a,b}\sqrt{\frac{\varepsilon_U^b}{\varepsilon_U^a}}\left[A_U^a \quad 0\right] \tag{6.44b}$$

and

$$\varepsilon_{U_a}^b = \varepsilon_U^b\left(1 - \left(\rho_U^{a,b}\right)^2\right). \tag{6.44c}$$

■

Proposition 6.4.2 shows that $B_{U_a}^b$ and $\varepsilon_{U_a}^b$ can be computed recursively. Substantial simplification in this recursion results when $\rho_U^{a,b} = 0$, corresponding to the maximum-entropy case. Indeed, in this case we have

$$B_{U_a}^b = \left[0 \quad B_U^b\right] \tag{6.45a}$$

and

$$\varepsilon_{U_a}^b = \varepsilon_U^b. \tag{6.45b}$$

Next, we consider the LLS estimate of z_a given $z_{U_b} = \left[z_U^T \quad z_b\right]^T$. The estimator has the form $L_{U_b}^a z_{U_b}$ where the row-vector $L_{U_b}^a$ satisfies the normal equations

$$\underbrace{\left[1 \quad -L_{U_b}^a\right]}_{A_{U_b}^a} P_{Q^2} = \left[\varepsilon_{U_b}^a \quad 0 \quad \cdots \quad 0\right] \tag{6.46}$$

where $\varepsilon^a_{U_b}$ is the estimation error variance and $A^a_{U_b}$ is as defined in (6.46). As the following proposition shows, we can also compute these quantities recursively.

Proposition 6.4.3. *Using the notation previously defined,*

$$\underbrace{\begin{bmatrix}1 & -L^a_{U_b}\end{bmatrix}}_{A^a_{U_b}} = \begin{bmatrix}A^a_U & 0\end{bmatrix} - \rho^{a,b}_U\sqrt{\frac{\varepsilon^a_U}{\varepsilon^b_U}}\begin{bmatrix}0 & B^b_U\end{bmatrix} \tag{6.47a}$$

and

$$\varepsilon^a_{U_b} = \varepsilon^a_U\left(1 - \left(\rho^{a,b}_U\right)^2\right). \tag{6.47b}$$

Proof. Using the augmented normal equations (6.40) we have

$$\left\{\begin{bmatrix}A^a_U & 0\end{bmatrix} - \frac{\delta^{a,b}_U}{\varepsilon^b_U}\begin{bmatrix}0 & B^b_U\end{bmatrix}\right\}P_{Q^2} = \begin{bmatrix}\varepsilon^a_U - \frac{\left(\delta^{a,b}_U\right)^2}{\varepsilon^b_U} & 0 & \cdots & 0\end{bmatrix} \tag{6.48a}$$

$$= \begin{bmatrix}\varepsilon^a_U\left(1 - \left(\rho^{a,b}_U\right)^2\right) & 0 & \cdots & 0\end{bmatrix}. \tag{6.48b}$$

Hence, we conclude that

$$\underbrace{\begin{bmatrix}1 & -L^a_{U_b}\end{bmatrix}}_{A^a_{U_b}} = \begin{bmatrix}A^a_U & 0\end{bmatrix} - \frac{\delta^{a,b}_U}{\varepsilon^b_U}\begin{bmatrix}0 & B^b_U\end{bmatrix} \tag{6.49a}$$

$$= \begin{bmatrix}A^a_U & 0\end{bmatrix} - \rho^{a,b}_U\sqrt{\frac{\varepsilon^a_U}{\varepsilon^b_U}}\begin{bmatrix}0 & B^b_U\end{bmatrix} \tag{6.49b}$$

and

$$\varepsilon^a_{U_b} = \varepsilon^a_U\left(1 - \left(\rho^{a,b}_U\right)^2\right). \tag{6.49c}$$

■

Again, in the maximum-entropy case for which $\rho^{a,b}_U = 0$, we have

$$A^a_{U_b} = \begin{bmatrix}A^a_U & 0\end{bmatrix} \tag{6.50a}$$

and

$$\varepsilon^a_{U_b} = \varepsilon^a_U. \tag{6.50b}$$

We now illustrate how the recursions of Proposition 6.4.2 and Proposition 6.4.3 may be applied to address covariance extension by continuing the example discussed previously.

Step	*Element*	*Quantities Computed*	*Can Recursively Compute*
1	$p_{a,b}$	$A^a_{\{u\}}, B^b_{\{u\}}, \varepsilon^a_{\{u\}}, \varepsilon^b_{\{u\}}, \rho^{a,b}_{\{u\}}$	$A^a_{\{u,b\}}, \varepsilon^a_{\{u,b\}}$
2	$p_{u,v}$	$A^u_{\{b\}}, B^v_{\{b\}}, \varepsilon^u_{\{b\}}, \varepsilon^v_{\{b\}}, \rho^{u,v}_{\{b\}}$	$B^v_{\{u,b\}}, \varepsilon^v_{\{u,b\}}$

Table 6.1. Summary of the first two one-element extensions of the partial covariance matrix of Figure 6.5.

Example: 4×4 Banded Partial Covariance Matrix, Continued

Consider again the partial covariance matrix illustrated in Figure 6.5 and the sequence of chordal graphs used to complete it as illustrated in Figure 6.6. On page 129 we discussed determination of the elements $p_{a,b}$ and $p_{u,v}$. The quantities computed in determining these covariance elements are listed in Table 6.1. We now discuss the final step—the computation of element $p_{a,v}$. To determine the value of this element we will rely on the recursions of Proposition 6.4.2 and Proposition 6.4.3. This final one-element extension step corresponds to adding edge "3" of Figure 6.6 to form the complete graph G^3. The maximal clique of G^3 is $\{a, u, b, v\} = \{a, v\} \cup (\{a, u, b\} \cap \{u, b, v\})$ where $\{a, u, b\}$ and $\{u, b, v\}$ are maximal cliques of G^2. Therefore, to apply (6.33) to determine $p_{a,v}$, we need to compute $A^a_{\{u,b\}}$, $B^v_{\{u,b\}}$, $\varepsilon^a_{\{u,b\}}$, and $\varepsilon^v_{\{u,b\}}$.

Consider first the quantities $B^v_{\{u,b\}}$ and $\varepsilon^v_{\{u,b\}}$. As indicated in Table 6.1, these are uniquely determined by certain previously determined quantities. Specifically, using Proposition 6.4.2, the quantities $B^v_{\{u,b\}}$ and $\varepsilon^v_{\{u,b\}}$ may be computed recursively from $A^u_{\{b\}}$, $B^v_{\{b\}}$, $\varepsilon^u_{\{b\}}$, $\varepsilon^v_{\{b\}}$, and $\rho^{u,v}_{\{b\}}$ all of which are available from the second one-element extension step. Similarly, using Proposition 6.4.3, the quantities $A^a_{\{u,b\}}$ and $\varepsilon^a_{\{u,b\}}$ may be computed recursively from $A^a_{\{u\}}$, $B^b_{\{u\}}$, $\varepsilon^a_{\{u\}}$, $\varepsilon^b_{\{u\}}$, and $\rho^{a,b}_{\{u\}}$ all of which are available from the first one-element extension step, as indicated in Table 6.1.

In the example just discussed, we were able to use the generalized-Levinson recursion to simplify covariance completion. Actually, the structure of this problem does not require the generality that the generalized-Levinson algorithm provides. In fact, in the case of computing a completion of a banded covariance matrix, the generalized-Levinson algorithm is equivalent to the classical one. We emphasize that the generalized-Levinson algorithm is applicable to a *much* wider range of extension problems. For instance, it is applicable to problems for which the initial known covariance data corresponds to any chordal graph, not just the case for which the initial known covariance data correspond to a banded pattern. The following example provides a simple extension problem that cannot be addressed with the classical Levinson algorithm but can be solved using our generalized algorithm.

Example: Simple Tree

Consider the partial covariance matrix illustrated in Figure 6.8. The known elements (indicated with filled circles) correspond to G^0, the simple tree shown in Figure 6.9 with dashed lines. Also shown in Figure 6.9 is the sequence of edges used to form

	d	a	b	c
d	●	●	●	●
a	●	●	?	?
b	●	?	●	?
c	●	?	?	●

Figure 6.8. Partial covariance matrix corresponding to the tree illustrated in Figure 6.9. Filled circles represent known elements, question marks represent unknown ones.

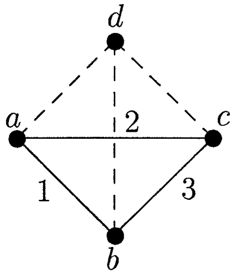


Figure 6.9. Chordal sequence for completing the matrix of Figure 6.8. The graph G^0 has edges indicated by dashed lines and is a tree. The edge sequence is indicated by the numbered labels: graph G^1 is formed by the addition of edge "1", graph G^2 by the addition of edge "2," and graph G^3 by the addition of edge "3."

the sequence of one-element extensions for this problem. Each edge corresponds to the determination of one of the unknown elements of the partial covariance matrix of Figure 6.8 (each unknown element is indicated with a question mark). The edge (a, b) (labeled "1") is added first, the edge (a, c) (labeled "2") is added second, and the edge (b, c) (labeled "3") is added last. The generalized-Levinson recursion can be applied to compute the quantities needed for $p_{b,c}$ (the last element to be computed) from previously computed quantities. However, this problem is outside the purview of the classical Levinson algorithm.

The generalized-Levinson algorithm, but not its classical counterpart, is also applicable to problems for which an extension to a pattern of entries corresponding to any chordal graph is sought rather than a full completion. The latter case arises in the context of building a MAR model for the maximum-entropy extension of a banded partial covariance matrix, a topic we develop in Section 6.5.

To compute the completion of a banded partial covariance matrix it is *always* possible to arrange the order of calculations to take advantage of the efficiency provided by the recursion of Levinson's algorithm. That is, it is always possible to choose a chordal sequence upon which to base the sequence of one-element extensions so that

New Edge	*New Max. Clique*
(a, d)	$\{a, d\}$
(c, d)	$\{c, d\}$
(c, e)	$\{c, e\}$
(b, e)	$\{b, e\}$
(a, c)	$\{a, c, d\}$
(b, c)	$\{c, b, e\}$
(a, b)	$\{a, b, c\}$

Table 6.2. Edges and maximal cliques for which the edge end-points are active corresponding to the chordal sequence indicated in Figure 6.3.

the quantities required for extension are also those provided by the Levinson recursion. For banded partial covariance matrices, determining unknown covariance elements one diagonal band at a time and working outward from the main diagonal toward the upper-right and lower-left corners *always* works. However, for a more general extension problem (one outside the purview of classical Levinson techniques) it is unclear whether such a chordal sequence exists. Indeed, it is possible to devise chordal sequences that do not allow one to make use of the generalized-Levinson recursion. An illustration of this is shown in the next example.

Example: Failed Recursion

Consider the sequence of chordal graphs indicated in Figure 6.3 in which the sequence of new edges is indicated with the integers 1–7. The sequence of new edges and the sequence of new maximal cliques for which the edge end-points are active are indicated in Table 6.2. Notice that, when edge (a, b) is a added, the new maximal clique is $\{a, b, c\} = \{a, b\} \cup (\{a, c\} \cap \{b, c\})$. Therefore, to perform this one-element extension step we must compute $A^a_{\{c\}}$, $B^b_{\{c\}}$, $\varepsilon^a_{\{c\}}$, $\varepsilon^b_{\{c\}}$. The quantities $A^a_{\{c\}}$ and $\varepsilon^a_{\{c\}}$ can, in principle, be computed recursively from $A^a_\emptyset$ and $B^c_\emptyset$. However, for the particular chordal sequence selected, these quantities are *not* available from previous computation. This is due to the fact that the maximal clique $\{a, c\}$ has not been formed in a previous graph. Indeed, $A^a_\emptyset$ and $B^c_\emptyset$ are only computed when the maximal clique $\{a, c\}$ is formed with a and c the active elements. Note that this does not mean that we cannot perform this extension step. It just means that we cannot rely on the generalized-Levinson recursion to do so. Instead, we must solve the necessary normal equations directly rather than recursively.

The previous example shows that, in order to make use of the generalized-Levinson recursion, we require that our chordal sequence possess an additional property. What is required is that the chordal sequence also be *efficient*, a property defined formally as follows.

Definition 6.4.1 (Efficient Chordal Sequence). *Let $[G^0, \ldots, G^n]$ be a sequence of chordal graphs with $G^i = (V, E^i)$ and $E^i = E^{i-1} \cup e^i$ where $e^i = (a^i, b^i) \notin E^{i-1}$. Let U^i a*

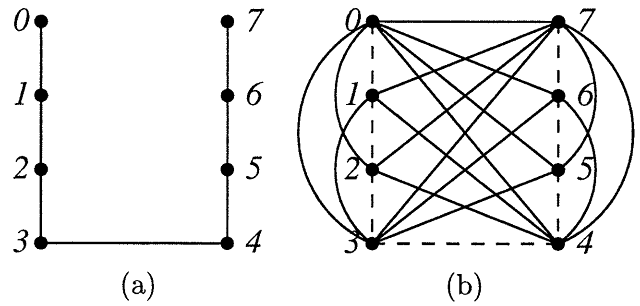


Figure 6.10. (a) $G^0 = (V, E^0)$ where E^0 is given by (6.51) for $N = 8$ and $k = 1$. (b) The edges added to G^0 to form G^n are those with solid lines.

clique of G^{i-1} such that the unique maximal clique of G^i containing a^i, b^i is $\{a^i, b^i\} \cup U^i$. Then this sequence of graphs is said to be an efficient chordal sequence *if, for some $j < i$, and $k < i$,*

(i) G^j has a maximal clique $\{a^i\} \cup U^i$ with a^i an active vertex, and

(ii) G^k has a maximal clique $\{b^i\} \cup U^i$ with b^i an active vertex.

Given an arbitrary choice of two chordal graphs $G^0 = (V, E^0)$ and $G^n = (V, E^n)$ such that $E^0 \subset E^n$, it is unclear (as of this writing) whether there exists an efficient chordal sequence between them. That not every chordal sequence is efficient was shown by the example illustrated in Figure 6.3 and Table 6.2. If the chordal sequence on which the order of one-element extensions are based is efficient, then the generalized-Levinson recursion may be used and the computational complexity of each step is, generally, linear in the cardinality of the set U. However, consideration of the maximum-entropy extension results in some simplification. In particular, since $\rho_U^{a,b} = 0$, the recursive computation of (6.42) and (6.47) corresponds to the augmentation of B_U^b and A_U^a by zero. Therefore, in the maximum-entropy case, the complexity of each one-element extension is a constant size, independent of the cardinality of the set U.

6.5 MAR Models for Maximum-Entropy Completions

We are now in a position to consider designing a MAR model for the maximum-entropy completion, P_{ME}, of the banded partial covariance matrix P_{E^0} where

$$E^0 = \{(m,n) \mid |m-n| \leq k\} \tag{6.51}$$

is the set of edges for graph $G^0 = (V, E^0)$ and $V = \{0, 1, \ldots, N-1\}$. For the case for which $N = 8$ and $k = 1$, the graph G^0 is depicted in Figure 6.10(a). As has been discussed, P_{ME} is the covariance matrix for a k-th order Markov process. Thus, when $N = 4k2^M$ for some integers k and M, P_{ME} can be modeled *exactly* by an $(M+1)$-scale end-point MAR model for a k-th order Markov process of the form described in

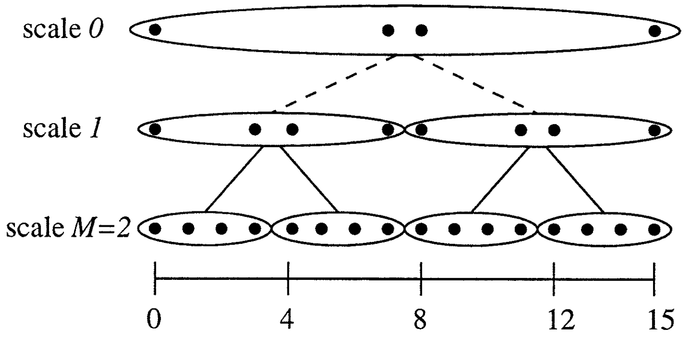


Figure 6.11. MAR model for a first-order Markov process: $M = 2$, $k = 1$.

Section 2.3.2. Thus, each internal matrix, W_s, is a selection matrix that picks out end-points of intervals. In particular, as discussed in Section 2.3.2, the indices selected by W_s are

$$\eta(s) = \eta_1(s) \cup \eta_2(s) \cup \eta_3(s) \tag{6.52a}$$

where

$$\eta_1(s) = \imath(s)4k2^{M-m(s)} + [0 : k-1], \tag{6.52b}$$
$$\eta_2(s) = \imath(s)4k2^{M-m(s)} + 4k2^{M-m(s)-1} + [-k : k-1], \tag{6.52c}$$
$$\eta_3(s) = \imath(s)4k2^{M-m(s)} + 4k2^{M-m(s)} + [-k : -1]. \tag{6.52d}$$

An example for the case of $M = 2$ and $k = 1$ is illustrated in Figure 6.11 (the significance of the dashed lines in this figure will be discussed in the sequel).

Having defined the internal matrices of our MAR model for P_{ME} as those of an end-point MAR model, we may use them to determine the joint statistics for child-parent pairs $(P_{x(s)}, P_{x(s\alpha_i)}, P_{x(s\alpha_i)x(s)})$ as discussed in Section 2.3 where P_{ME} plays the role of P_{fM}. In turn, the state covariances and child-parent cross-covariances are used to determine the MAR parameters as in (3.7). Therefore, it would seem that the heart of the problem is to compute P_{ME}. However, as we will show, only $O(N)$ elements out of a total of N^2 elements in P_{ME} are actually needed. We point out that the fact that the elements in each state are a subset of those in its children states (i.e., $\eta(s) \subset \eta(s\alpha_1) \cup \eta(s\alpha_2)$) implies that our end-point MAR model is internal.

To compute the joint child-parent statistics for node s, we need only the elements of P_{ME} indexed by $\mathcal{C}_s \triangleq \eta(s) \cup \eta(s\bar{\gamma})$. This holds for all $s \in \mathcal{S}_0 - \{0\}$. Therefore, we need not compute the maximum-entropy completion of P_{E^0}. Rather, we require an extension to P_{E^n} where

$$E^n = \bigcup_{s \neq 0} \mathcal{C}_s \times \mathcal{C}_s. \tag{6.53}$$

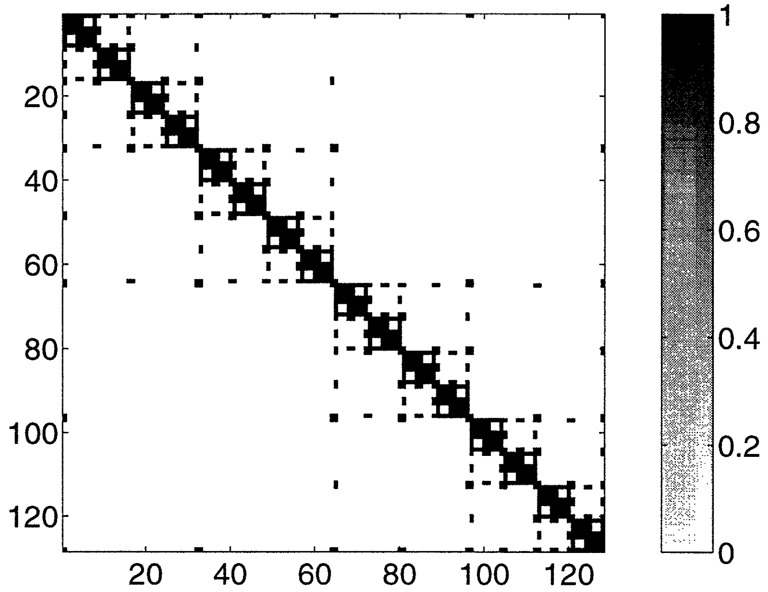


Figure 6.12. The adjacency matrix for $G^n = (V, E^n)$ when $k = 1$, $M = 5$. Black indicates the subset of entries needed for P_{E^n}.

The graph $G^n = (V, E^n)$ is illustrated in Figure 6.10(b) for the case in which $M = 1$ and $k = 1$ (i.e., a first-order AR process with $N = 8$ points). The dashed lines are edges of E^0, and the solid lines are edges of $E^n - E^0$. We illustrate G^n to show that, while it possesses considerable structure, it is also quite complex. These facts are also clear from the adjacency matrix for $G^n = (V, E^n)$, illustrated in Figure 6.12. This matrix indicates (in black) the subset of the elements of P_{ME} that are in P_{E^n} for the case for which $k = 1$ and $M = 5$ (i.e., a first-order AR process with $N = 128$ points). Figure 6.12 clearly shows that there are a vast number of entries of P_{ME} that are not needed to build a MAR model. We now make this statement precise. Since $|\eta(s)| = 4k$ we have that $|\mathcal{C}_s| \leq 8k$ and $|\mathcal{C}_s \times \mathcal{C}_s| \leq 64k^2$. These are upper bounds because the sets $\eta(s)$ as well as the sets $\mathcal{C}_s \times \mathcal{C}_s$ are not mutually exclusive, which can be seen by inspection of Figure 6.11. Therefore, the total number of elements in E^n is bounded above by $64k^2|\mathcal{S}_0| = 64k^2(2^{M+1} - 1)$. Finally, since $N = 4k2^M$, we have that $|E^n| < 32kN$. While this is an upper bound, it is of the right order and it indicates that we require only $O(N)$ elements of P_{ME}.

We now discuss the application of the generalized-Levinson algorithm of Section 6.4 to compute P_{E^n}, the required elements of P_{ME}. For the moment, assume that (1) $E^0 \subset E^n$, and (2) $G^n = (V, E^n)$ is chordal. We will prove these two facts shortly. Assuming these, there exists a chordal sequence from G^0 to G^n. We may, therefore, find the elements of P_{E^n} using the generalized-Levinson algorithm of Section 6.4 by setting the generalized reflection coefficients to zero. We now show that doing so results in $O(N)$ computational complexity even when the chordal sequence upon which we base

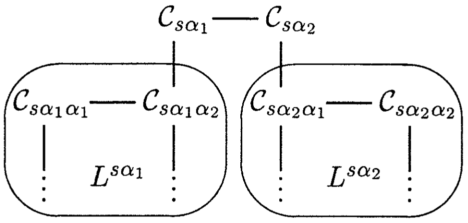

$\mathcal{C}_{s\alpha_1}$
$\mathcal{C}_{s\alpha_2}$
$\mathcal{C}_{s\alpha_1\alpha_1}$
$\mathcal{C}_{s\alpha_1\alpha_2}$
$\mathcal{C}_{s\alpha_2\alpha_1}$
$\mathcal{C}_{s\alpha_2\alpha_2}$
$L^{s\alpha_1}$
$L^{s\alpha_2}$

Figure 6.13. The graph L^s.

our one-element extensions is not efficient. Recall that if our chordal sequence is not efficient we must solve some (in the worst case, all) of the normal equations that arise in the generalized-Levinson algorithm explicitly, rather than recursively. However, the largest maximal clique of G^n has cardinality $|\mathcal{C}_s| < 8k$. Hence, at worst, each one-element extension requires $O(k^3)$ computations. This leads to an overall complexity of $O(N)$ for computing P_{E^n} under the assumption that k is independent of N. The complexity of computing P_{E^n} using an efficient chordal sequence is also $O(N)$ so there is no computational advantage (asymptotically) in using an efficient sequence. In the sequel we do not assume that our chordal sequence is efficient. Note that the preceding discussion shows that the complexity of computing a MAR model for P_{ME} is of the same order as computing an AR model using the classical Levinson algorithm (in the nonstationary case).

We now turn to the facts we assumed in the previous paragraph: (1) $E^0 \subset E^n$, and (2) $G^n = (V, E^n)$ is chordal. First, we address (1). The following proposition shows that $E^0 \subset E^n$, and the proof is based on the fact that the states of the end-point MAR model collectively contain all intervals of length $k+1$.

Proposition 6.5.1. *$E^0 \subset E^n$ where these are defined by* (6.51) *and* (6.53).

Proof. The proof follows from the fact that, by construction, every interval of length $k+1$ is contained in one of the sets $\mathcal{C}_s$. This is most easily seen by considering Figure 6.11 which represents a MAR model for a first-order Markov process ($k = 1$, $M = 2$). It is clear from this figure that every interval of length 2 can be found in some $\mathcal{C}_s$. The algebraic details proving this fact for a general k and M are provided in Appendix D. ■

The next task is to show that G^n is chordal. We will show this by exhibiting a junction tree for G^n and using Proposition 6.2.2. For these purposes, we define L^s to be the graph with vertex set $\text{vert}(L^s) = \{\mathcal{C}_t \mid t \in \mathcal{S}_s - \{s\}\}$ and edge set

$$\text{edge}(L^s) = \{(\mathcal{C}_{t\alpha_1}, \mathcal{C}_{t\alpha_2}) \mid t \in \mathcal{S}_s\} \cup \{(\mathcal{C}_{t\alpha_1}, \mathcal{C}_{t\alpha_1\alpha_2}) \mid t \in \mathcal{S}_s\} \cup \{(\mathcal{C}_{t\alpha_2}, \mathcal{C}_{t\alpha_2\alpha_1}) \mid t \in \mathcal{S}_s\}. \tag{6.54}$$

An illustration of L^s is provided in Figure 6.13. Notice that L^s has as subgraphs $L^{s\alpha_1}$ and $L^{s\alpha_2}$.

Proposition 6.5.2. *G^n is chordal.*

Proof. L^0 is a junction tree for G^n. This fact can be seen intuitively from Figure 6.13 which, when specialized to the case of $s = 0$, is clearly a tree on the set of maximal cliques of G^n, namely $\{\mathcal{C}_s\}$. A rigorous proof of this fact is provided in Appendix D. The proposition then follows from Proposition 6.2.2. ■

While we have completed the formal description of how to build a MAR model for P_{ME} there is one remaining algorithmic detail—that of finding a chordal sequence between G^0 and G^n. As mentioned in Section 6.2, this may always be done by brute force search. However, the structure of the particular problem at hand suggests a more elegant approach to finding a chordal sequence that includes a subsequence of chordal graphs with an appealing scale-recursive structure.

Consider, for $n \in [1 : M]$, the graph given by $H^n = (V, K^n)$ where

$$K^n \triangleq E^0 \bigcup_{s:m(s)\geq M-n+1} \mathcal{C}_s \times \mathcal{C}_s\,. \tag{6.55}$$

Therefore, with the sequence $\Gamma \triangleq [G^0, H^1, H^2 \ldots , H^M = G^n]$ we are building up from G^0 to G^n by adding the necessary cliques scale-recursively. The recursion begins by adding edges corresponding to covariance elements for finest-scale statistics. With each successive graph H^n, edges for covariance elements at the next coarser scale are added. For example, referring to Figure 6.11, the child-parent joint statistics

$$\begin{pmatrix} P_{x(s)} & P_{x(s)x(s\bar{\gamma})} \\ P^T_{x(s)x(s\bar{\gamma})} & P_{x(s\bar{\gamma})} \end{pmatrix} \tag{6.56}$$

are first computed for all child-parent pairs linked by solid lines. Then, proceeding to the next coarser scale, the joint child-parent statistics are computed for the pairs linked by dashed lines. By showing that H^n is chordal (which we do in the next proposition), we are assured by Proposition 6.2.1 that there is a chordal sequence that contains Γ as a subsequence.

Proposition 6.5.3. *$H^n = (V, K^n)$ where K^n is defined in (6.55) is chordal.*

Proof. See Appendix D. ■

While H^n is chordal, Γ is not a chordal sequence because the transition from H^{n-1} to H^n involves more than one edge. However, having decomposed the problem scale-recursively, finding a sequence of edges to transition from H^{n-1} to H^n while maintaining chordality is not hard. In fact a chordal sequence may be obtained with the property that, for a given scale $m(s) = m(t)$, the edges of $\mathcal{C}_s$ are added prior to those of $\mathcal{C}_t$ for all $s \neq t$ such that $\imath(s) < \imath(t)$. This ordering of the sets $\{\mathcal{C}_r\}_{r\in\mathcal{T}_n(0)}$ corresponds to deducing the statistics $P_{x(s)}$, $P_{x(s)x(s\bar{\gamma})}$, $P_{x(s\bar{\gamma})}$ prior to $P_{x(t)}$, $P_{x(t)x(t\bar{\gamma})}$, $P_{x(t\bar{\gamma})}$.

Proposition 6.5.4. *Using the notation previously defined, there exists a chordal sequence that contains* $[G^0, H^1, H^2 \ldots, H^M = G^n]$ *as a subsequence with the property that the transition from* H^{n-1} *to* H^n *is accomplished in such a way that for all* s *and* t *such that* $m(s) = m(t) = M - n + 1$ *and* $\imath(s) < \imath(t)$*, the edges of* C_s *are added prior to those of* C_t*.*

While a formal proof of Proposition 6.5.4 is not provided in this thesis, a sketch is found in Appendix D. We also point out that this fact is easy to check by computer for any particular case (i.e., any choice of k and M).

In this section we have shown how to build efficiently (with a complexity linear in problem size) a MAR model for the maximum-entropy completion of a banded partial covariance matrix. Our approach is based on the generalized-Levinson algorithm which, in turn, relies on a sequence of chordal graphs. We have exhibited a particular sequence of chordal graphs and we emphasize that the computational complexity of our approach *does not* require that this sequence be efficient. It is unclear how to build efficiently MAR models for other (non-maximum-entropy) completions. The reason is that it is unclear what the state definitions ought to be. In the development presented in this section, we were able to select our states to be those corresponding to an end-point MAR model *precisely* because we sought the maximum-entropy extension (which, as we showed, corresponds to a Markov process). This problem of state definition must be addressed if one is to generalize the approach developed in this chapter.

Chapter 7

Incorporation of Nonlocal Variables

IN this chapter we extend the realization framework developed in previous chapters to accommodate estimation problems involving nonlocal variables. We present three different approaches to incorporating nonlocal variables into a MAR model all of which differ in important ways from the approach discussed in Section 2.3.5 [37,39]. These approaches represent a different modeling philosophy than adopted in preceding chapters. In preceding chapters we have considered a MAR model as an implicit and (possibly) approximate representation of a covariance matrix, P_{fM}, for a fine-scale random process. In this chapter we take a substantially different point of view and we consider a MAR model to be a means of implicitly and exactly or approximately representing an estimator. As we will show, for the purposes of well-approximating an estimator, in many cases it is neither sufficient nor is it necessary to well-approximate P_{fM}.

The exclusive focus of the preceding chapters on the finest scale is reasonable if one is only interested in producing fine-scale sample-paths or one is interested only in fine-scale estimates based on a reasonably dense set of fine-scale measurements. However, when one is interested in estimating or fusing multiresolution variables, this is not a justifiable approach for several reasons. First, the realization algorithms of preceding chapters cannot accommodate arbitrary nonlocal variables. That is, a given nonlocal variable cannot generally be represented as a linear function of the elements of a single state in the realized model (recall our discussion of this point in Section 2.3.5). Indeed, to accommodate nonlocal variables, another constraint (in addition to internality and Markovianity) must be imposed: internal matrices must include specific linear functionals that represent nonlocal measurements or variables to estimate. This new constraint can be accommodated approximately (as is done in Section 7.1 using what we shall call the *approximate nonlocal method*) or exactly (as is done in Section 7.2 using what we shall call the *exact nonlocal method*).

Another reason why focusing on the finest scale is not appropriate in some multiresolution estimation problems is that, in doing so, the resulting models may allocate resources (i.e., state dimension and computation) in capturing fine-scale statistical features that are irrelevant for the estimation problem at hand. The degree to which resources are ill-spent depends on details of the estimation problem (i.e., the measurement geometry and the particular estimates of interest). Roughly speaking, the source of this poor allocation of resources is that the Markov property is overkill and using it risks trying to conditionally decorrelate variables that are of no relevance to estimation. As we show in Section 7.3, in designing a MAR state $x(s)$ for an estimation problem, one need only consider the conditional decorrelation of the (in some cases, many fewer) variables that represent data or estimates that are separated by s. This idea leads to what we shall call *goal directed modeling.*

Throughout this chapter we consider the following type of LLS estimation problem. Let f^M be a vector representing a fine-scale signal or (lexicographically ordered) image. Let g be the vector of data which is related to f^M by

$$g = Hf^M + \nu \tag{7.1}$$

where P_ν is diagonal and ν is uncorrelated with f^M. We are interested in the LLS estimate not of f^M, but of the linear function of f^M given by Df^M. So that we may evaluate the quality of our estimates, we are also interested in the estimation error variances $P_e(i,i)$ (i.e., the diagonal elements of P_e) where $e = Df^M - \widehat{\mathrm{E}}[Df^M \mid g]$. If D and H both consist of rows of the identity then we have a fine-scale estimation problem with point-wise measurements, which is a case that is adequately handled by the theory and algorithms of preceding chapters. Therefore, we will consider cases in which one or both of D and H contain nonlocal linear functionals (i.e., one or both have rows which contain more than one nonzero element). The LLS estimation equations for this problem are

$$\begin{aligned}\widehat{\mathrm{E}}[Df^M|g] &= P_{Df^M,g}(P_{Hf^M} + P_\nu)^{-1}g && (7.2a)\\ &= DP_{f^M}H^T(HP_{f^M}H^T + P_\nu)^{-1}g && (7.2b)\end{aligned}$$

and

$$\begin{aligned}P_e(i,i) &= [P_{Df^M}]_{i,i} - [P_{Df^M,g}(P_{Hf^M} + P_\nu)^{-1}P^T_{Df^M,g}]_{i,i} && (7.3a)\\ &= [DP_{f^M}D^T]_{i,i} - [DP_{f^M}H^T(HP_{f^M}H^T + P_\nu)^{-1}HP_{f^M}D^T]_{i,i}\,. && (7.3b)\end{aligned}$$

To well-approximate an estimator, a MAR model must well-approximate the matrices that make up (7.2) and (7.3), namely $P_{Df^M,g}$, P_{Hf^M}, P_ν, and $[P_{Df^M}]_{i,i}$. Doing so places an additional constraint on the internal matrices, as mentioned, and dealing with this constraint is the subject of this chapter.

In addition to the notation just introduced, we also define L_s to be a matrix such that

$$W_s f_s^M = L_s f^M\,. \tag{7.4}$$

That is, L_s contains the internal matrix W_s as a submatrix and is appropriately zero-padded so that right multiplication by f^M makes sense. This is identical to the definition of the linear functionals $\{L_s\}$ given in Section 2.3.

7.1 Approximate Representation of Nonlocal Variables

In this section we assume that an internal and, possibly, approximate MAR model, $x(\cdot)$, for P_{f^M} has already been determined. Then, we seek to accommodate the nonlocal variables associated with a particular estimation problem. If nonlocal variables are to be incorporated exactly, state augmentation is required as discussed in Section 2.3.5 (see also [37,39]). This increase in state dimension leads to less efficient signal processing algorithms. As an alternative, in this section we consider incorporating nonlocal variables *approximately* and in such a way that results in *no* increase in state dimension.

To begin, recall that the MAR measurement equation (cf., (2.18)) is

$$y(s) = C(s)x(s) + v(s) \tag{7.5}$$

where $v(\cdot)$ is white and $v(s)$ has covariance $R(s)$. We now discuss the problem of reconciling (7.1) with (7.5) so that nonlocal measurements may be incorporated in the MAR model. Consider the i-th row of (7.1) which corresponds to a scalar nonlocal measurement of the form

$$g_i = h_i^T f^M + \nu_i \tag{7.6}$$

where it is assumed that h_i^T is a nonlocal linear functional. If there is node s such that h_i^T is in the row-space of L_s then g_i can be represented exactly as, say, the j-th measurement[1] at node s by

$$g_i = y_j(s) = c_j(s)^T x(s) + v_j(s) \tag{7.7}$$

where $c_j(s)^T$ is the j-th row of $C(s)$ and $c_j(s)^T L_s = h_i^T$. Typically, however, there will be no s such that h_i^T is in the row-space of L_s. However, there is some node s such that h_i^T is closest to the row-space of L_s in a mean-square error sense. Therefore, the measurement g_i can be approximately captured as follows. Let $\widehat{g}_i^s = \widehat{\mathrm{E}}[h_i^T f^M \mid L_s f^M]$ be the LLS estimate of g_i using the information contained at node s. Then, in representing g_i in terms of $\widehat{g}_i^s$, there are two sources of error, a measurement error ν_i and a modeling error $\widetilde{g}_i^s \triangleq h_i^T f^M - \widehat{g}_i^s$. That is,

$$g_i = \widehat{g}_i^s + \nu_i + \widetilde{g}_i^s \tag{7.8a}$$

where

$$\mathrm{var}(\widetilde{g}_i^s) = h_i^T P_{f^M} h_i - c_j(s)^T P_{f^M} c_j(s) \tag{7.8b}$$

and

$$c_j(s)^T = h_i^T P_{f^M} L_s^T (L_s P_{f^M} L_s^T)^{-1} . \tag{7.8c}$$

[1] There may be more than one measurement at a given node.

Thus, the optimal node at which to incorporate measurement g_i is given by

$$t_i = \arg\min_{s \in \mathcal{S}_0}\bigl(\mathrm{var}(\widetilde{g}_i^s)\bigr)\,. \tag{7.9}$$

Suppose now that we have placed all measurements (all elements of g) optimally[2] using (7.9). Consider any node s that contains measurements. The rows of $C(s)$ are given by (7.8c) and we now consider the measurement error covariance $R(s)$. If the measurements at node s are captured exactly (i.e., modeling error $\widetilde{g}_i^s$ is zero for all g_i placed at node s) then the only measurement error stems from ν which is white. Thus $R(s)$ is diagonal with the diagonal elements (the variances) given by the corresponding ones from P_ν. However, for those measurements at node s that are only approximately captured, the modeling errors $\widetilde{g}_i^s$ are nonzero and possibly correlated with one another. Therefore, the diagonal elements of $R(s)$ are of the form $\mathrm{var}(\nu_i) + \mathrm{var}(\widetilde{g}_i^s)$ where the first term comes fro P_ν and the second from (7.8b). Note that the variables ν_i and $\widetilde{g}_i^s$ are uncorrelated because the former is uncorrelated with f^M and the latter is a linear function of f^M by definition.

We now turn to the off-diagonal terms of $R(s)$. These are readily computed by a generalization of (7.8b) to the vector case. To this end, let $\widetilde{g}^s$ be the vector of modeling errors associated with measurements placed at node s. That is, $\widetilde{g}^s$ consists of several $\widetilde{g}_i^s$. Let H_s be the matrix whose rows consist of the corresponding h_i^T. Then

$$P_{\widetilde{g}^s} = H_s P_{f^M} H_s^T - C(s) P_{f^M} C(s)^T\,. \tag{7.10}$$

The off-diagonal terms of $R(s)$ are equivalent to the off-diagonal terms of $P_{\widetilde{g}^s}$ and account for the fact that the modeling errors for the measurements placed at node s are correlated. Of course, the modeling errors are also correlated across nodes. That is, the modeling errors $\widetilde{g}^s$ at node s are correlated with those at node t, namely $\widetilde{g}^t$. This cross-correlation is *not* captured by our model and this fact represents another source of error.

This completes the specification of the approximate representation of nonlocal measurements in the MAR framework. In the sequel, we will refer to the technique developed in this section as the *approximate nonlocal (AN) method.* The complexity of the AN method when used to incorporate a single nonlocal measurement g_i depends on the supports of h_i^T, L_s, and $c_j(s)^T$. In the worst case, the complexity is $O(N^2)$ where N is the length of f^M. This complexity stems from the need to compute matrix-vector products like $P_{f^M} h_i$ (cf., (7.8b)). The complexity is, at best, $O(N)$ per nonlocal measurement because (7.9) is a search over all $O(N)$ nodes.

We now turn to the approximate representation of nonlocal estimates. That is, consider the LLS estimate of variable $z = d_i^T f^M$ where d_i^T is a nonlocal linear functional

[2] We emphasize that placing measurements optimally using the criterion of (7.9) does not imply that the model is good. The ultimate measure of the quality of the model is the degree to which the resulting estimates and error variances deviate from those which are obtained by solving the normal equations using the exact statistics.

contained in the i-th row of D. Assuming that the MAR model well-approximates f^M, a good approximation to the exact LLS estimate of z may always be obtained based on $\widehat{x}^M$ which is provided by the MAR estimator. That is, using the fact that $\widehat{x}^M \approx \widehat{f}^M$ we have $\widehat{z} \approx d_i^T \widehat{x}^M$. Efficiently computing a good approximation to the estimation error variance $\text{var}(z - \widehat{z})$ is a more difficult issue. The reason is that the MAR estimator provides estimation error covariances for each $x(s)$ but not the global estimation error covariance for all states collectively. Therefore, when d_i^T is in the row-space of L_s for some s, the error variance is readily computed as $d_i^T P_{e(s)} d_i$ where the quantity $P_{e(s)}$ is the covariance for the estimation error $e(s) = x(s) - \widehat{x}(s)$ at node s and is provided by the MAR estimator. However, when d_i^T is not in the row-space of any L_s, computing an approximate error variance is more difficult since it involves off-diagonal blocks of the global estimation error covariance. The off-diagonal blocks corresponding to states close in tree distance can be computed with a only a little extra work using the MAR model for the estimation error (which is also provided by the MAR estimator [127, 131]). Therefore, using the MAR error model to compute some (but not all) of the off-diagonal blocks, it may be feasible to compute an approximate error variance for $\widehat{z}$. Clearly there are several open issues associated with this idea, namely which off-diagonal blocks to compute and how to bound the modeling error induced by ignoring the ones not computed.

■ 7.2 Exact Representation of Nonlocal Variables

In contrast to the previous section, in this one we consider capturing nonlocal linear functionals exactly and in a manner similar in spirit to the method of [37, 39] and discussed in Section 2.3.5. The difference is that in [37, 39], states of an existing model are augmented to incorporate linear functionals whereas the approach of this section is to incorporate linear functionals first and then build a model around them. There are several advantages to the latter approach. First, in building a model around previously incorporated linear functionals, we may condition on the information they carry. Consequently, there is no redundancy and we avoid the singularity problems that arise using the approach of [37, 39]. Second, because we use the information carried by nonlocal linear functionals to aid in the decorrelation of random variables, our states are lower dimensional while fulfilling the same roles (decorrelation and representation of nonlocal linear functionals) as those of derived using the method of [37,39]. Third, our approach provides a natural and computationally efficient way to build exact or approximate models that are always internal. In contrast, the approach of [37,39] must begin with an exact model.

Our procedure begins with an $(M + 1)$-scale q-adic tree commensurate with the length N of signal f^M. We will represent nonlocal linear functionals at nodes of this tree, however, in contrast to all previously discussed methods, we do not first define internal matrices. Consider a nonlocal linear functional of the form $b^T f^M$ which may represent a measurement or a variable to be estimated. The first step is to decide at what node to represent b^T. While a procedure for choosing the best node (in some sense) is elusive,[3] let us assume that (somehow) we have selected a node s at which to represent b^T.

The second step is to enforce explicitly our desire to be able to represent b^T at node s. This is done in a manner similar to the procedure described in Section 2.3.5. That is, we will reserve q rows of the internal matrix L_s (which is not yet defined) in which to place pieces of b^T. More specifically, rows j through $j+q-1$ of L_s will contain the matrix[4]

$$G_s = \begin{pmatrix} b_{s\alpha_1}^T & 0 & \cdots & 0 \\ 0 & b_{s\alpha_2}^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & b_{s\alpha_q}^T \end{pmatrix} \tag{7.11}$$

where

$$b^T f^M = \sum_{i=1}^{q} b_{s\alpha_i}^T f_{s\alpha_i}^M . \tag{7.12}$$

Notice that in breaking up the linear functional b^T into its components in this way, we are forcing L_s to be block diagonal. In turn this ensures that, ultimately, we end up with an internal model (cf., Proposition 4.1.1).

Having placed b^T at node s, we must ensure that the information contained in b is propagated to the leaf nodes so that each state element is a linear function of its children states. This is done in a manner similar to the procedure just described for adding b. Specifically, we are going to add $b_{s\alpha_i}$ to $L_{s\alpha_i}$ by defining a submatrix, $G_{s\alpha_i}$, of $L_{s\alpha_i}$. Each row of $G_{s\alpha_i}$ will contain the component of $b_{s\alpha_i}$ that acts on the states indexed by the descendents of one of the children of $s\alpha_i$. Continuing this process scale-recursively until the finest scale is reached will ensure that our model is internal and that the linear functionals we have added are consistent with the fine-scale process ultimately defined by our MAR model.

To add several nonlocal linear functionals we may repeat the foregoing procedure. After doing so, the measurement model (7.5) is easy to specify since every measurement is, by construction, captured at some node exactly. Additionally, every nonlocal variable to estimate is represented exactly. The last step is to build a MAR model around the linear functionals previously added. This is done by applying either of the procedures of

[3] What is known is that the freedom in choosing a node is constrained by internality. For internality to hold, b^T can only be captured at node s if $b^T f^M = a^T f_s^M$ for some vector a. Thus, every linear functional can be represented at the root node and, generally, a linear functional can be represented at more than one node. The only other criteria are that, ultimately, we want a model that is low-dimensional and does a good job of approximating the Markov property. However, how to translate these into a formal procedure for choosing a node at which to represent a linear functional is unclear.

[4] If b is the first linear functional to be added at node s then $j = 1$. In general, however, other linear functionals have already been added to node s.

Chapter 4 (the $O(N^2)$ one or the $O(N)$ boundary approximation) with the additional step that, when computing a given local internal matrix at node s, we first *condition* the relevant random quantities on the information already present at node s due to the prior incorporation of the nonlocal linear functionals. Therefore, the additional information added for the purposes of approximately or exactly achieving Markovianity is not redundant. In the sequel, we will refer to the technique developed in this section as the *exact nonlocal (EN) method.*

If the number of nodes at which nonlocal linear functionals (or parts thereof required for internality) are incorporated using the EN method is independent of N then the overall asymptotic complexity of the EN procedure is $O(N^2)$ or $O(N)$ where, in the latter case, the boundary approximation is used. If, however, nonlocal linear functionals are incorporated at a number of nodes proportional to N then the computational complexity is, at worst, $O(N^3)$ or, for the boundary approximation, $O(N^2)$. This worst-case complexity arises, for example, if *every* node represents a nonlocal linear functional (or a part thereof as required for internality). To see this, consider the problem of designing a local internal matrix at node s at scale $n-1$. In doing so, we will need to conditionally decorrelate variables f^n at the preceding finer scale, n, *after* conditioning on the information carried by the nonlocal linear functionals at s. Suppose, for example, that $b_s^T f^M$ is represented at node s. Then, conditioning f^n on $b_s^T f^M$ requires computing the *conditional* correlation matrix $P_{f^n|b_s^T f^M}$. This matrix has $O(d^2 q^{2n})$ entries so to compute it requires an amount of work proportional to $O(d^2 q^{2n})$. To design the local internal matrix at each node, a *different* conditional correlation matrix must be computed because the nonlocal information carried by each node is different. Summing this over all nodes in the tree leads to an $O(N^3)$ complexity. In the case of the boundary approximation, however, only $O(dq^n)$ entries of $P_{f^n|b_s^T f^M}$ need to be computed, leading to an $O(N^2)$ complexity.

■ 7.3 Markovianity and Estimation

When considering a MAR model as an implicit representation of an estimator involving nonlocal variables, the Markov property is neither necessary nor is it sufficient to achieve exact estimates and error variances. In this section we will elaborate on this point and provide an intellectual successor to the Markov property that is more appropriate for model identification in the context of addressing a particular estimation problem. Our focus is on the quantities relevant for the estimation problem of (7.2) and (7.3), namely

$$P_{Df^M,g} = DP_{f^M}H^T\,, \tag{7.13a}$$

$$P_{Hf^M} = HP_{f^M}H^T\,, \tag{7.13b}$$

$$P_\nu\,, \tag{7.13c}$$

$$[P_{Df^M}]_{i,i} = [DP_{f^M}D^T]_{i,i}\,. \tag{7.13d}$$

To build a MAR model for the estimation problem of (7.2) and (7.3) we need to capture the four quantities (7.13a)–(7.13d). As described in the Section 7.2, we can represent exactly the linear functionals given by the rows of D and H in our model by explicitly incorporating them. A consequence of this is that $[DP_{f^M}D^T]_{i,i} = d_i^T P_{f^M} d_i$ can be captured exactly by some state of the MAR model.[5] Therefore (7.13d) is accommodated. Similarly, because each measurement is captured exactly we have that, for each i, $[P_\nu]_{i,i} = [R(s)]_{j,j}$ some node s and index j. Therefore (7.13c) is also accommodated.

It remains only to accommodate (7.13a) and (7.13b). The former corresponds to the cross-covariance between the variables to be estimated and the noise-free measurements. The latter corresponds to the covariance for the noise-free measurements. Since the rows of D and H are exactly represented by states in the model, exactly capturing (7.13a) and (7.13b) corresponds to ensuring that the cross correlations among certain states are exact. In particular, capturing (7.13a) exactly requires the cross-correlations between all states with variables to estimate and all states with measurements to be captured exactly. Similarly, capturing exactly (7.13b) requires the cross-correlations between all pairs of states with measurements to be captured exactly. The crux of the modeling problem, therefore, is, given two states $x(s)$ and $x(t)$ at specific nodes s and t, to ensure that $\mathrm{E}[x(t)x(s)^T] = L_t P_{f^M} L_s^T$. As we will see, this condition can be satisfied with states of lower dimension than is required to achieve an exact model for f^M by designing the internal matrices that define $x(\cdot)$ appropriately.

To develop the procedure for achieving $\mathrm{E}[x(t)x(s)^T] = L_t P_{f^M} L_s^T$, we first consider a tree-indexed process $f(\cdot)$ whose fine-scale covariance is P_{f^M} defined by

$$f(s\alpha_i) = A(s\alpha_i)f(s) + \mu(s\alpha_i)\,, \tag{7.14a}$$

$$Q(s\alpha_i) \triangleq \mathrm{E}[\mu(s\alpha_i)\mu(s\alpha_i)^T] \tag{7.14b}$$

where, as in previous chapters, $\mu(\cdot)$ is *not* necessarily a white noise process. The parameters $A(s\alpha_i)$ and $Q(s\alpha_i)$ are computed from $P_{f(s)}$ and $P_{f(s\alpha_i)f(s)}$ which are, themselves, defined by P_{f^M}, L_s, $L_{s\alpha_i}$ via

$$P_{f(s)} \triangleq L_s P_{f^M} L_s^T\,, \tag{7.15a}$$

$$P_{f(s\alpha_i)f(s)} \triangleq L_{s\alpha_i} P_{f^M} L_s^T \tag{7.15b}$$

where we have used the fact that $f(s) = L_s f^M$. As described in Section 4.1, if $\mu(\cdot)$ is a white noise process then (7.14) is a MAR process. However, in the event that $\mu(\cdot)$ is not a white noise process then we can *define* a MAR process as

$$x(s\alpha_i) = A(s\alpha_i)x(s) + w(s\alpha_i)\,, \tag{7.16a}$$

$$\mathrm{E}[w(s\alpha_i)w(s\alpha_i)^T] = Q(s\alpha_i) \tag{7.16b}$$

[5] Recall from Chapter 4 that, even for approximate MAR models, state covariances and child-parent cross-covariances are exact.

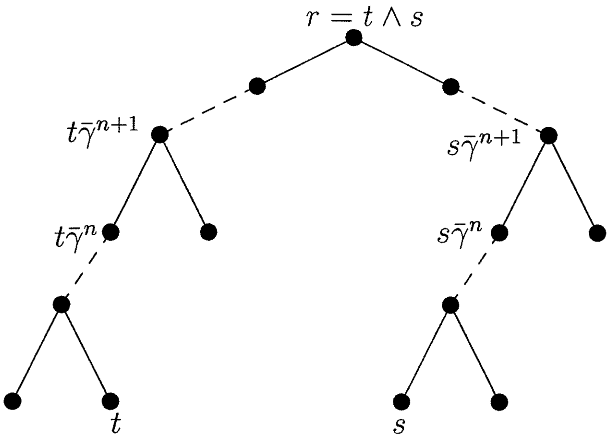


Figure 7.1. A visual interpretation of the relationship among the variables of Proposition 7.3.1. Nodes t and s reside at the same scale and r is their common ancestor of maximal scale. A dashed line indicates a path through the tree connecting the line's end-points.

where $A(\cdot)$ and $Q(\cdot)$ are the same as in (7.14) and $w(\cdot)$ is white, uncorrelated with $x(0)$. As discussed in Section 4.1, the state covariances $P_{x(s)}$ at each node s and the child-parent cross-covariances $P_{x(s\alpha_i)x(s)}$ for each child-parent pair of nodes exactly match $P_{f(s)}$ and $P_{f(s\alpha_i)f(s)}$, respectively.

We now turn to the conditions that the $\{L_s\}$ must satisfy so that $\mathrm{E}[x(t)x(s)^T] = L_t P_{f^M} L_s^T$. In our development of these we will rely on the following lemma.

Lemma 7.3.1. *Conditioned on G_0z the vectors G_1z and G_2z are uncorrelated if and only if*

$$G_1 P_z G_2^T = G_1 P_z G_0^T (G_0 P_z G_0^T)^{-1} G_0 P_z G_2^T \,. \tag{7.17}$$

Proof. By definition, G_0z conditionally decorrelates G_1z and G_2z if

$$\mathrm{E}\left[\left(G_1z - \widehat{\mathrm{E}}[G_1z \mid G_0z]\right)\left(G_2z - \widehat{\mathrm{E}}[G_2z \mid G_0z]\right)^T\right] = 0 \,. \tag{7.18}$$

The proof follows by expanding the expression under the expectation and taking the expected value of the resulting terms. ■

Using Lemma 7.3.1, we first provide the sufficient conditions so that that $\mathrm{E}[x(t)x(s)^T] = L_t P_{f^M} L_s^T$ for the case where s and t are at the same scale. For a visual interpretation of the relationship among the variables of Proposition 7.3.1 see Figure 7.1.

Proposition 7.3.1. *Let $x(\cdot)$ be an internal MAR process defined by (7.16). That is, the parameters of the process $x(\cdot)$ are the ones derived for the process $f(\cdot)$ defined by (7.14). For $t, s \in \mathcal{S}_0$ suppose $m(t) = m(s) \triangleq m$. Let $r = t \wedge s$ be the common ancestor*

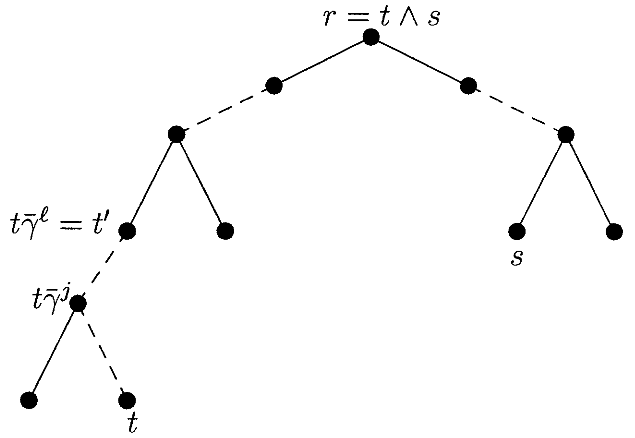


Figure 7.2. A visual interpretation of the relationship among the variables in Proposition 7.3.2 for which $m(t) > m(s)$. A dashed line indicates a path through the tree connecting the line's end-points.

of t and s with maximal scale. Then, $\mathrm{E}[x(t)x(s)^T] = L_t P_{fM} L_s^T$ if the process $f(\cdot)$, has the property that for $n \in \{m - m(r) - 1, m - m(r) - 2, \ldots, 0\}$,

$$f(t\bar{\gamma}^n) \text{ is uncorrelated with } \begin{bmatrix} f(s\bar{\gamma}^n) \\ f(s\bar{\gamma}^{n+1}) \end{bmatrix} \text{ when conditioned on } f(t\bar{\gamma}^{n+1}), \text{ and} \tag{7.19a}$$

$$f(s\bar{\gamma}^n) \text{ is uncorrelated with } \begin{bmatrix} f(t\bar{\gamma}^n) \\ f(t\bar{\gamma}^{n+1}) \end{bmatrix} \text{ when conditioned on } f(s\bar{\gamma}^{n+1}). \tag{7.19b}$$

Proof. See Appendix E. ■

Next, we provide the sufficient conditions such that $\mathrm{E}[x(t)x(s)^T] = L_t P_{fM} L_s^T$ for the case where s and t are at different scales. Figure 7.2 provides a visual interpretation of the relationships among the variables of Proposition 7.3.2.

Proposition 7.3.2. *Let $x(\cdot)$ and $f(\cdot)$ be as defined in Proposition 7.3.1. Suppose for $t, s \in \mathcal{S}_0$, $m(t) > m(s)$. Let $t' = t\bar{\gamma}^{m(t)-m(s)}$ and $r = t \wedge s$, the common ancestor of t and s with maximal scale. Then, $\mathrm{E}[x(t)x(s)^T] = L_t P_{fM} L_s^T$ if the process $f(\cdot)$ has the property that, for $n \in \{m(s) - m(r) - 1, m(s) - m(r) - 2, \ldots, 0\}$,*

$$f(t'\bar{\gamma}^n) \text{ is uncorrelated with } \begin{bmatrix} f(s\bar{\gamma}^n) \\ f(s\bar{\gamma}^{n+1}) \end{bmatrix} \text{ when conditioned on } f(t'\bar{\gamma}^{n+1}), \text{ and} \tag{7.20a}$$

$$f(s\bar{\gamma}^n) \text{ is uncorrelated with } \begin{bmatrix} f(t'\bar{\gamma}^n) \\ f(t'\bar{\gamma}^{n+1}) \end{bmatrix} \text{ when conditioned on } f(s\bar{\gamma}^{n+1}) \tag{7.20b}$$

and also, for $j \in \{m(t) - m(s) - 1, m(t) - m(s) - 2, \ldots, 0\}$,

$$f(t\bar{\gamma}^j) \text{ is uncorrelated with } f(s) \text{ when conditioned on } f(t\bar{\gamma}^{j+1}). \tag{7.21}$$

Proof. See Appendix E. ■

While the algorithm of Chapter 4 focuses on the scale-recursive Markov property, Proposition 7.3.1 and Proposition 7.3.2 suggest a slightly different notion of Markovianity. Indeed, in designing the local internal matrix at node s, rather than focus on every state at the next finer scale (as the scale-recursive Markov property does), we need only consider the subset of states that carry information about variables to estimate or measurements. In the remainder of this section, we will make this idea precise as we develop an algorithm based on the conditional decorrelation conditions of Proposition 7.3.1 and Proposition 7.3.2. This algorithm will produce an internal and exact or approximate model that incorporates nonlocal linear functionals associated with any particular estimation problem. Since it exploits the structure of the given estimation problem, it provides a better allocation of both state dimension and computational resources.

The description of the algorithm for designing a MAR model $x(\cdot)$ for an estimation problem involving nonlocal variables is simplified by labeling the nodes of the tree that indexes $x(\cdot)$. There are six kinds of node labels and any node may have any number (including none) of them. An unlabeled node will be called an *empty* node. The labels are e, ec, ep, m, mc, and mp which denote, respectively:

- estimate (e) node,
- estimate consistency (ec) node,
- estimate path (ep) node,
- measurement (m) node,
- measurement consistency (mc) node,
- measurement path (mp) node.

A node s is labeled e (m) if a variable to estimate (a measurement) is placed at s, which is done as described in Section 7.2. A node is denoted ep (mp) if it is along the path from a node labeled e (m) to the root node. An example is illustrated in Figure 7.3 in which a variable to estimate and a measurement are represented at nodes 2 and 4, respectively. Notice that all nodes along the path from 2 to 0 and from 4 to 0 are labeled ep and mp, respectively. A node s is labeled ec (mc) if both for some n, $s\bar{\gamma}^n$ is labeled e (m) and also at least one descendent of $s\bar{\gamma}$ is labeled either e or m. Therefore, no nodes in the tree in Figure 7.3 are labeled ec or mc. Consider, however, Figure 7.4, in which the node labels of Figure 7.3 have been augmented to indicate that a variable to estimate appears at node 1. Thus, nodes 3 and 4 are labeled[6] ec because $3\bar{\gamma} = 4\bar{\gamma} = 1$ is an e node and a descendent of $3\bar{\gamma} = 4\bar{\gamma}$ (namely, node 4) is an m node. No other node

[6] As will be explained, it is not necessary that node 3 be an ec node. However, the subsequent development is simplified if every node either has its full complement of children labeled or has none of them labeled.

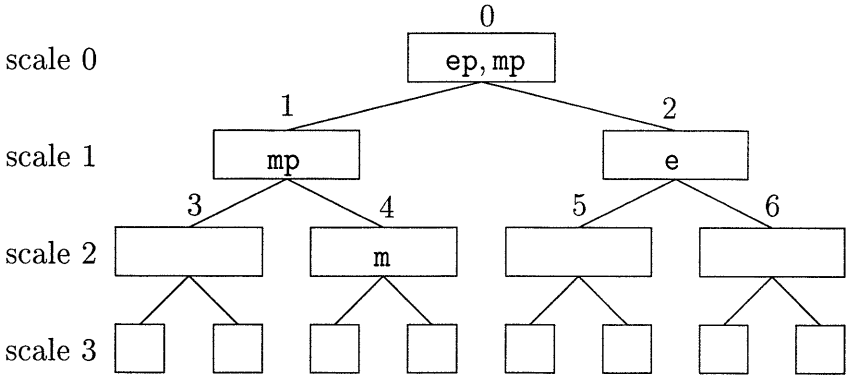


Figure 7.3. An example of e, ep, m, and mp labels. A variable to estimate and a measurement are represented at nodes 2 and 4, respectively.

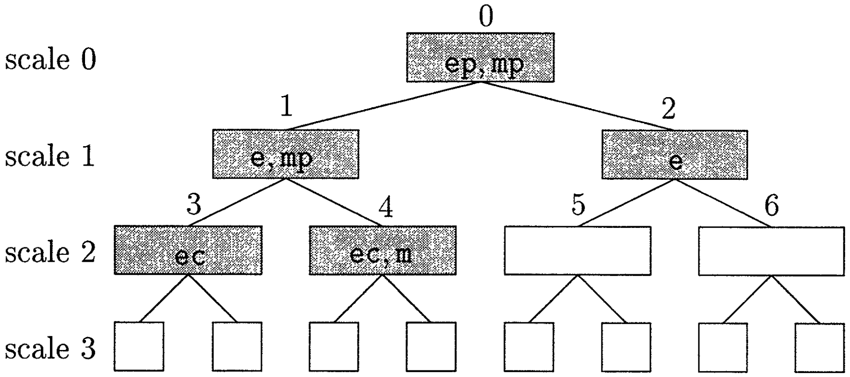


Figure 7.4. Nodes 3 and 4 are labeled ec because 1 is labeled e and 4 is labeled m (see text). The shaded nodes represent the nodes in the pruned tree described in the text.

satisfies the criterion for an ec or mc label. For instance, while node 5 does have an ancestor (at node 2) that is labeled e, there are no descendents of $5\bar{\gamma}$ that are labeled e or m.

Consider building a MAR model for an estimation problem with the characteristics implied by Figure 7.4. That is, there is one (nonlocal) measurement characterized by a linear functional we shall call h which has been placed at node 4. There are two (nonlocal) variables to estimate which have been placed at nodes 1 and 2 which are characterized by linear functionals d_1 and d_2, respectively. To arrive at an internal and consistent model, we must augment states descending from e and m nodes so that the information they carry is propagated down the tree (as described in Section 7.2). However, there is no need to propagate information to all descending nodes because many of them will play no role in the estimation problem. That is, the empty (unlabeled) nodes of Figure 7.4 are roots of subtrees which contain no measurements and no variables of interest to estimate. These nodes may, therefore, be removed and we arrive at a pruned tree which contains only the shaded nodes of Figure 7.4. It is clear from this pruned tree that, for consistency, only nodes 3 and 4 need to carry information regarding a coarser-scale variable (namely, the d_1 linear functional at node 1). Note that the information at node 3 exists only so that the d_1 can be written in terms of variables at nodes 3 and 4. That is, the propagation of part of d_1 to node 4 is required so that it is consistent with h (as described in Section 7.2). The other part of d_1 is propagated to node 3. While this latter propagation is *not* necessary, our development is simplified if every node either has its full complement of children labeled or has none of them labeled.

Using the node labels of Figure 7.4, we may design states to capture exactly the cross-correlations between the measurement (at node 4) and the variables to estimate (at nodes 1 and 2) as follows. The internal matrices at nodes 2, 3, and 4 are already completely defined since they contain linear functionals corresponding to measurements and variables to estimate or parts of linear functionals needed for consistency (as is indicated by the labels e, ec, and m). All of these linear functionals are built into the internal matrices at nodes 2, 3, and 4 as described in Section 7.2. Consider designing the internal matrix for node 1. Some of the rows are already defined so as to capture d_1^T, which represents the variable to estimate at node 1 (indicated by the label e). Since we seek to capture the cross-correlations between the measurement and the variables to estimate exactly, the role of the variables at node 1 is to conditionally decorrelate the variables at node 4 (which contains a measurement) and those of node 2 (which contains a variable to estimate). This is a pair-wise decorrelation problem which is easily solved using the techniques of Section 2.4 (after conditioning on the information present at node 1 as described in Section 7.2). Notice that there is no need to conditionally decorrelate the variables at node 3 and node 2 since those of node 3 play no role in the estimation problem other than providing internality. Now consider the root node. The variables at the root node must conditionally decorrelate the measurement at node 4 from the variable to estimate at node 2. To do so, it is sufficient to satisfy the conditions of Proposition 7.3.2 (cf., (7.21)) and, thus, we must conditionally decorrelate the variables at nodes 1 and 2. This too is a pair-wise decorrelation problem.

We may generalize the procedure illustrated by the preceding example as follows. The first step is to choose a q-adic tree with $M+1$ scales commensurate with the length of f^M. The second step is to choose nodes at which to represent measurements and variables to estimate. These are the m nodes and the e nodes. The mp, ep, mc, ec, and empty nodes are also defined by this choice. The empty nodes may be pruned from the tree as they will play no role in defining the model. The internal matrices of the leaf nodes (those with no children) of this pruned tree are already defined as they contain linear functionals (or parts thereof) corresponding to nonlocal measurements and variables to estimate. Next, we proceed scale-recursively,[7] beginning with scale $M-1$ and followed by scale $M-2$, etc. For each scale, we visit each node at that scale

[7] We still think of the pruned tree as having nodes organized into scales where the scale of node s of the pruned tree is the same as in the original (unpruned) one.

and determine the local internal matrix for that node. The order in which the nodes at a given scale are visited is irrelevant. It suffices to describe the procedure for defining the internal matrix for an arbitrary node.

To describe this procedure we will make use of the following notation. Let $\widetilde{\mathcal{S}}_0$ denote the set of nodes in the pruned tree while, as always, $\mathcal{S}_0$ is the set of nodes in the original regular tree. Similarly, using the notation of Section 2.2, we define

$$\begin{aligned}\widetilde{\mathcal{S}}_s &\triangleq \widetilde{\mathcal{S}}_0 \cap \mathcal{S}_s = \text{nodes in subtree of pruned tree rooted at } s\,,\\ \widetilde{\mathcal{T}}_s(n) &\triangleq \{t \in \widetilde{\mathcal{S}}_s \mid m(t) = n\} = \text{nodes of pruned tree at scale } n \text{ descending from } s\,.\end{aligned}$$

Consider the set of ancestors of node s in the original (unpruned) tree,

$$\mathcal{A}_s = \{t \in \mathcal{S}_0 \mid t = s\bar{\gamma}^{\ell} \text{ for some } \ell \geq 0\}\,, \tag{7.22}$$

and the subset of the set of ancestors that remain after pruning,

$$\widetilde{\mathcal{A}}_s = \mathcal{A}_s \cap \widetilde{\mathcal{S}}_0\,. \tag{7.23}$$

There is a unique node in $\widetilde{\mathcal{A}}_s$ of maximal scale. Denote this node by $\widehat{s}$ which is defined formally as

$$\widehat{s} = \arg\max_{t \in \widetilde{\mathcal{A}}_s} m(t)\,. \tag{7.24}$$

For example, referring to the tree of Figure 7.4, $\widehat{5} = 2$ because the node in the pruned tree which corresponds to the ancestor (in the original tree) of 5 with maximal scale is 2. Similarly, if s is a descendent of 5 (or, for that matter, 6) in the original tree $\widehat{s} = 2$ as well. Next, define

$$\widetilde{\mathcal{U}}_s(n) \triangleq \widetilde{\mathcal{T}}_s(n) \cup \{\widehat{t} \mid t \in \mathcal{T}_s(n) - \widetilde{\mathcal{T}}_s(n)\} \tag{7.25}$$

where $\mathcal{T}_s(n)$ is the subset of nodes of $\mathcal{S}_s$ that reside at scale n (as defined in Section 2.2). Basically, $\widetilde{\mathcal{U}}_s(n)$ is the set of nodes at scale n that descend from s in the pruned tree with the modification that, if a node t at scale n has been pruned away, the node of the pruned tree that is its ancestor (in the original tree) of maximal scale, $\widehat{t}$, is included in $\widetilde{\mathcal{U}}_s(n)$. Thus, considering again Figure 7.4, $\widetilde{\mathcal{U}}_0(2) = \{3, 4, 2\}$. Finally, define the following subsets of $\widetilde{\mathcal{U}}_0(n)$:

$$\widetilde{\mathcal{H}}_s(n) \triangleq \{t \in \widetilde{\mathcal{U}}_0(n) \mid t \text{ is labeled } \mathtt{m},\ \mathtt{mc}, \text{ or } \mathtt{mp}\}\,, \tag{7.26a}$$

$$\widetilde{\mathcal{D}}_s(n) \triangleq \{t \in \widetilde{\mathcal{U}}_0(n) \mid t \text{ is labeled } \mathtt{e},\ \mathtt{ec}, \text{ or } \mathtt{ep}\}\,. \tag{7.26b}$$

Returning now to the procedure for defining the local internal matrix at node s which resides at scale n, we proceed in a fashion similar to that of Section 7.2. That is, we are going to build a local internal matrix for node s by considering several pair-wise decorrelation problems, one for each child of s. For each such problem, we are going first to condition the relevant random variables on the linear functionals that have been placed at s. The solutions to these pair-wise problems are then concatenated as in (2.66). The structure of a particular pair-wise problem depends on the label of the node $s\alpha_i$ we are considering. If $s\alpha_i$ is an `m` or `mp` node (e.g., node 4 in the tree of Figure 7.4) then it indexes a measurement or is an ancestor of a node that does. Therefore, one of the roles of the state at node s is to conditionally decorrelate all of the measurements that reside at or descend from $s\alpha_i$ from all other measurements and variables to estimate in the tree so as to capture exactly (7.13a) and (7.13b). Sufficient conditions for achieving this decorrelation are given in the statements of Proposition 7.3.1 and Proposition 7.3.2. Since we are designing an internal model (in which each state is a linear function of its children states), these conditions further simplify. Indeed, we need only consider the pair-wise problem of conditionally decorrelating the variables at $s\alpha_i$ from all other variables at scale $n+1$ pertaining to measurements and variables to estimate, namely those indexed by $\widetilde{\mathcal{H}}_s(n+1) \cup \widetilde{\mathcal{D}}_s(n+1) - \{s\alpha_i\}$. For example, referring again to Figure 7.4, it is sufficient that the variables at 1 conditionally decorrelate the variables at node 4 (an `m` node) from the variables at nodes 2 (an `e` node) and 3 (an `ec` node), both of which belong to the set $\widetilde{\mathcal{D}}_1(2)$.

If, on the other hand, $s\alpha_i$ is an `e` or `ep` node (e.g., node 2 in the tree of Figure 7.4) then the role of $x(s)$ is to conditionally decorrelate all of the variables to estimate that reside at or descend from $s\alpha_i$ from all measurements so as to capture exactly (7.13a). Therefore, we need to consider the pair-wise problem of conditionally decorrelating the variables at $s\alpha_i$ from those indexed by $\widetilde{\mathcal{H}}_s(n+1) - \{s\alpha_i\}$. For example, referring again to Figure 7.4, the variables at node 0 need to conditionally decorrelate the variables at node 2 from those at node 1 which belongs to $\widetilde{\mathcal{H}}_0(1)$. Notice that the asymmetry in this procedure mimics the asymmetry of estimation. That is, measurements and variables to be estimated are *not* treated equivalently because, in solving an estimation problem, information flows from measurements to estimated variables but not vice versa. In the sequel, we will refer to the technique developed in this section as *goal directed modeling* (GDM).

The computational complexity of the foregoing procedure depends critically on the number, type (`e` or `m`), and node placement of nonlocal linear functionals. Clearly the complexity of this procedure is no greater than the one described in Section 7.2 and is typically smaller for two reasons. First, some nodes are pruned away and are, therefore, never visited in model realization. Second, as described in the previous paragraph, while variables at an `m` node must be conditionally decorrelated from many variables (those relating to both measurements and variables to estimate), variables at an `e` node must only be conditionally decorrelated from a smaller set of variables (those relating to measurements). Thus, the marginal computational cost of accommodating an additional `e` node is smaller than that of accommodating an additional `m` node. However, precise characterization of the computational complexity as a function of the number, type, and placement of nonlocal linear functionals is an open problem to which we will return in Chapter 8.

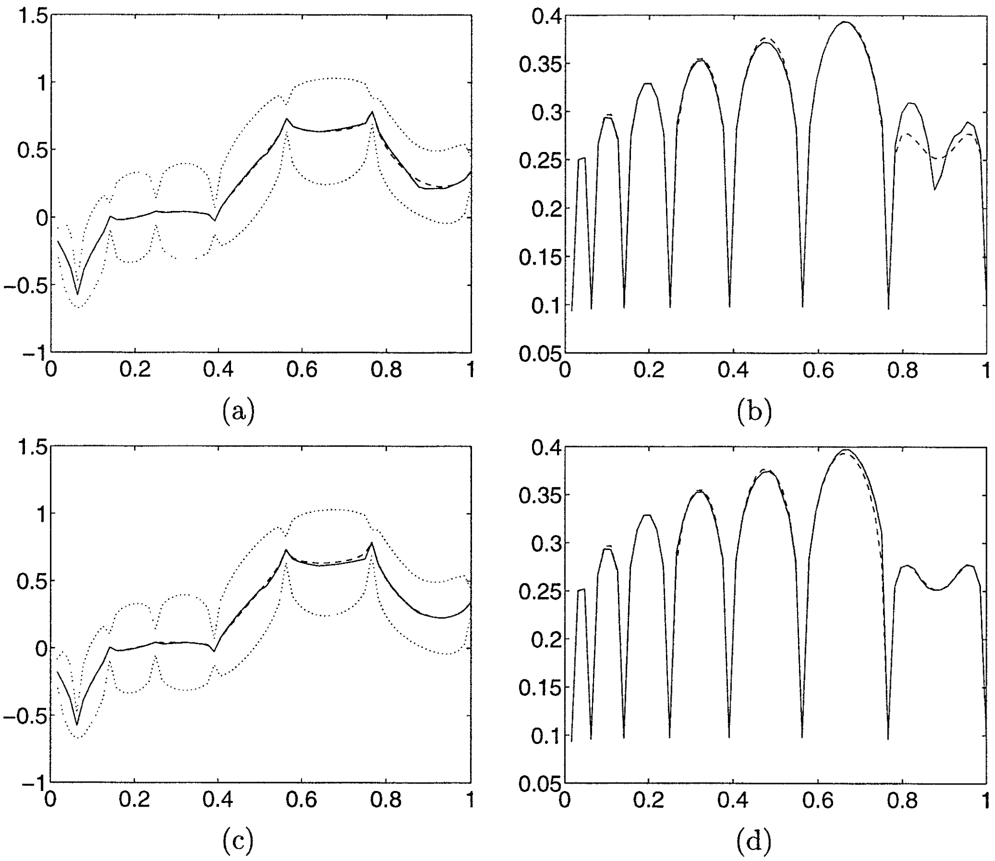


Figure 7.5. (a) MAR (solid line) and optimal (dashed line) estimates of fBm(0.3) based on sparse point-wise measurements and the sum over (3/4, 1], approximately incorporated using the AN method ($d = 4$). Plus/minus one standard deviation error bars (dotted lines) are also shown. (b) MAR error standard deviations (solid line) and optimal ones (dashed line). (c) and (d) represent processing similar to (a) and (b) but the nonlocal measurement is incorporated exactly using the EN method rather than approximately.

7.4 Examples

In this section we provide several examples illustrating the algorithms described in this chapter.

7.4.1 One-Dimensional Examples

In our first example we illustrate the AN method of Section 7.1 and the EN method of Section 7.2 and show that accurate estimation results can be obtained even when nonlocal measurements are approximated. Figure 7.5 illustrates the estimation of 64 samples of fBm(0.3) over (0, 1] from nine noisy measurements. Eight of the measurements are local point measurements of the samples indexed by perfect squares in [1 : 64]. This corresponds to measurements at times $t = 0.02, 0.06, 0.14, 0.25, 0.39, 0.56, 0.77, 1.00$. The ninth measurement is a nonlocal one representing the sum over the quarter-interval $(3/4, 1]$. For all measurements, the measurement noise has variance 0.01. The MAR model to be used to solve this estimation problem has state dimension $d = 4$ and we first consider a model for which the nonlocal measurement has been added approximately using the AN method. Specifically, the MAR process used as a model for this problem is indexed by a dyadic tree with five scales. The point-wise measurements are mapped to the finest scale, as has been done in the estimation problems considered in previous chapters. The single nonlocal measurement (the sum over $(3/4, 1]$) is represented at scale $m(s) = 2$ and shift $\imath(s) = 4$ which is, in some sense, the most natural node because the fine-scale descendents of this node correspond to the interval $(3/4, 1]$. Figure 7.5(a) illustrates the MAR estimates (solid line), optimal estimates based on exact statistics (dashed line) and one standard deviation error bars (dotted line). The largest difference between the exact and MAR estimates is in the interval $(3/4, 1]$—precisely the region associated with the approximately represented nonlocal functional. However, the MAR estimates are close to the optimal ones and well within the error bars indicating that the impact of the modeling error (due both to the approximate MAR model of fBm(0.3) and the approximate incorporation of the nonlocal linear functional) for this estimation problem is not significant. Figure 7.5(b) illustrates the MAR estimation error standard deviations (solid line) and the optimal ones (dashed line).

Next, we continue to consider the estimation problem described in the previous paragraph but we apply a MAR model (again, with state dimension $d = 4$ and indexed by a dyadic tree with five scales) for which the nonlocal measurement is added exactly using the EN method. Specifically, first the nonlocal linear functional is incorporated at scale $m(s) = 2$ and shift $\imath(s) = 4$ and the information carried by it is propagated down the tree as described previously. Then, a model is built around the information contained in node $(m(s), \imath(s)) = (2, 4)$ and its descendents by conditioning on it as described in Section 7.2. Figure 7.5(c) and Figure 7.5(d) illustrate estimates and error standard deviations associated with this model (solid lines) as well as the optimal ones (dashed lines). Notice that the estimates of Figure 7.5(c) are better (i.e., closer to the optimal ones) than those of Figure 7.5(a). This is most easily seen in the region $(3/4, 1]$. Additionally, the error standard deviations of the MAR model depicted in Figure 7.5(d) are noticeably closer to the optimal ones than are those of Figure 7.5(b). Again, this is most evident in the region $(3/4, 1]$.

In our second example, we compare the EN method of Section 7.2 to the GDM method of Section 7.3 for the problem of estimating 64 samples of a 15-th order stationary Markov process. There are three measurements in this problem: a point measurement at location 32, the sum of the four samples $[1 : 4]$ and the sum of the four samples $[61 : 64]$. The measurement noise has variance 0.1 for all three measurements. Figure 7.6(a) illustrates MAR (solid line) and optimal (dashed line) estimates based on a model with state dimension $d = 3$ designed using the EN method in which the linear functionals representing the two nonlocal measurements are incorporated exactly.

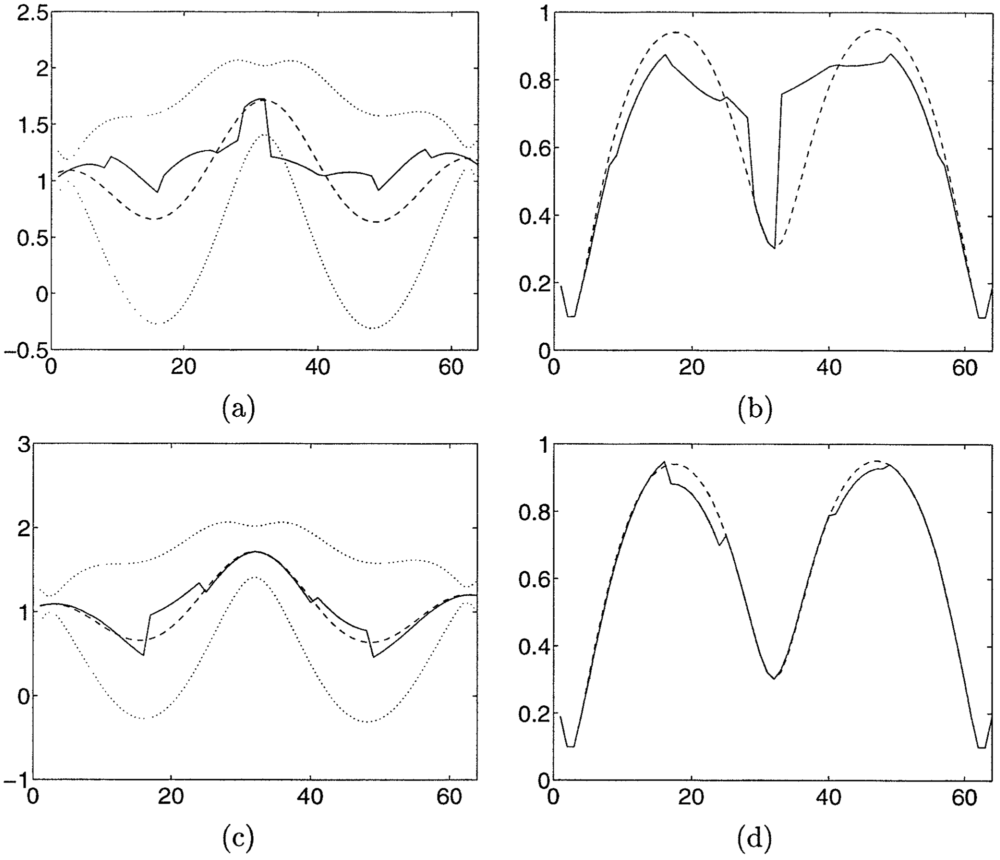


Figure 7.6. (a) MAR (solid line) and optimal (dashed line) estimates of a 15-th order stationary Markov process based on one point measurement (sample 32) and two nonlocal ones (sums of first and last four points) incorporated exactly using the EN method ($d = 3$). Plus/minus one standard deviation error bars (dotted lines) are also shown. (b) MAR error standard deviations (solid line) and optimal ones (dashed line). (c) and (d) represent processing similar to (a) and (b) but the model is built using the GDM procedure.

Plus/minus one standard deviation error bars (dotted line) are also shown. As explained in Section 7.2, the technique for computing the model parameters is based on the procedure of Section 4.1 (with the difference that we first condition on the nonlocal variables). The procedure of Section 4.1 relies on scale-recursive Markovianity which, as we will illustrate, is overkill for this problem because it focuses on variables that play no role in estimation. Notice that the MAR estimates are poor approximations to the optimal ones, with the maximum difference larger than one standard deviation. MAR (solid line) and optimal (dashed line) error standard deviations are shown in Figure 7.6(b). This plot illustrates that the MAR error standard deviations are also crude approximations to the optimal ones.

Figure 7.6(c) and Figure 7.6(d) illustrate estimates and error standard deviations,

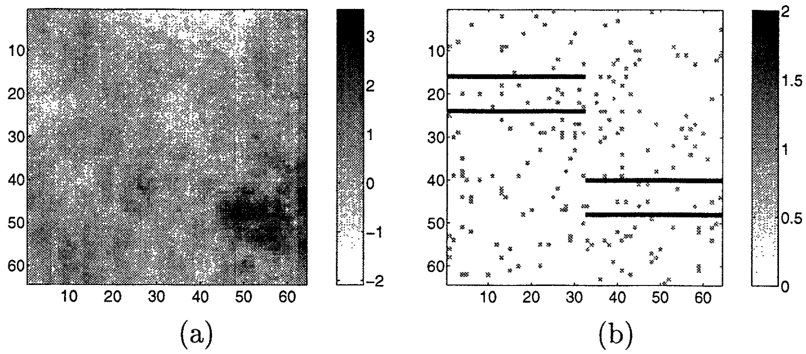


Figure 7.7. (a) Sample-path based on the statistics of (4.17). (b) Measurement geometry: point measurements (grey), line-integral measurements (black).

respectively, derived from a different MAR model designed using the GDM technique (solid line) and based on the optimal statistics (dashed line). The MAR model has state dimension $d = 3$. The GDM technique is not based on the scale-recursive Markov property and exploits the structure of the estimation problem. It focuses its attention only on conditionally decorrelating variables that *do* play a role in estimation and, therefore, achieves a higher fidelity model with no increase in state dimension. The improvement in estimates and error standard deviations is clear from the figures. This improvement is due to the fact that the GDM technique focuses state dimension resources on the estimation problem at hand and does not waste them on modeling parts of P_{f_M} that are not relevant to this problem.

■ 7.4.2 Two-Dimensional Multiresolution Data Fusion Example

In the next example, we illustrate a random field data fusion problem based on irregular local and nonlocal measurements. In particular, we estimate the field illustrated in Figure 7.7(a) based on measurements whose locations are indicated in Figure 7.7(b). The sample-path in Figure 7.7(a) is identical to the one depicted in Figure 4.10(a) and is drawn from the statistics of (4.17). Each grey point in Figure 7.7(b) corresponds to a point measurement. Only about 20% of the fine-scale pixels are measured and, as can be seen, the point measurements are scattered irregularly. The four horizontal black lines correspond to nonlocal sums. That is, we have four line-integral measurements taken over the regions indicated with black lines. The measurement noise variance is 0.2 for all measurements. Applying a MAR model designed using the EN method (and with $d = 64$) to the data fusion problem at hand, we obtain the MAR estimates and estimation error variances shown in Figure 7.8(a) and Figure 7.8(b), respectively. Figure 7.8(c) and Figure 7.8(d) illustrate estimates and error variances for the same estimation problem but associated with a MAR model with state dimension $d = 32$. Notice that the estimates of Figure 7.8(c) are nearly identical to those of Figure 7.8(a). Additionally, the estimation error variances of Figure 7.8(d) are on the same order as

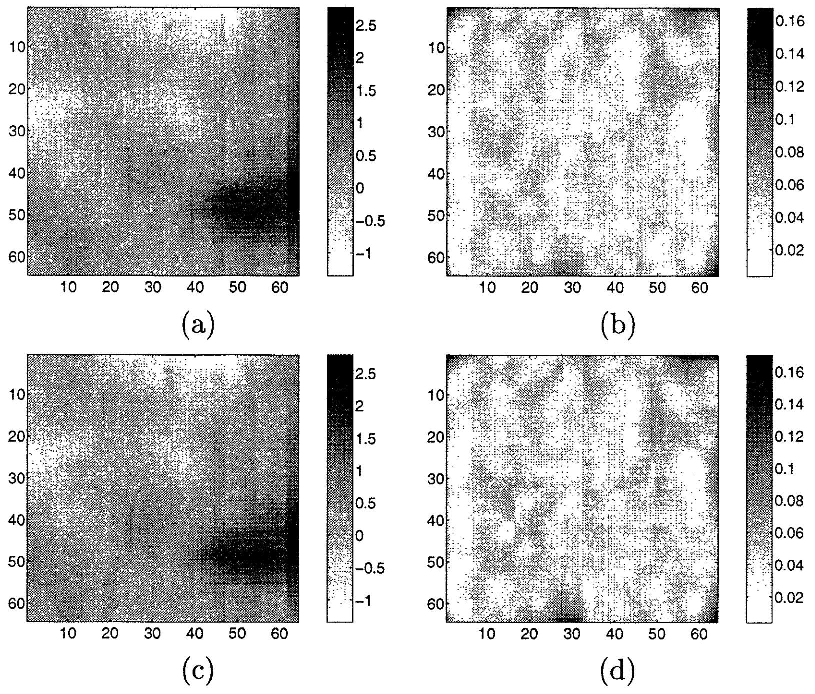


Figure 7.8. Random field multiresolution data fusion example. (a) Estimates based on the measurement geometry of Figure 7.7(b) using a MAR model designed using the EN method (state dimension $d = 64$). (b) Error variances. (c) and (d) represent processing similar to (b) and (c) but with a MAR model with state dimension $d = 32$.

those of Figure 7.8(b), and some blocky artifacts can be seen in the former due to the lower dimensionality of the model.

The estimates and error variances for a data fusion problem similar to the one just discussed are illustrated in Figure 7.9. The data for this case are identical to those for the previous example. However, in this case we are interested in viewing the problem at a coarser scale. That is, rather than estimating exclusively fine-scale pixels, we instead wish to design a MAR model which provides coarser scale estimates and error variances. Specifically, we build a MAR model that includes the average over 4×4 regions tiling the underlying 64×64 field. Additionally, we have used the GDM method in which each of these nonlocal linear functionals[8] is placed in the quad-tree at the unique node with maximal scale with the property that the support of the linear functional is contained in the fine-scale region (abstractly) represented at that node. Figure 7.9(a) illustrates the estimates for the 4×4 coarse-scale pixels based on a MAR model with state dimension $d = 64$ while Figure 7.9(b) illustrates the corresponding estimation error variances. Figure 7.9(c) and Figure 7.9(d) illustrate the 4×4 coarse-scale pixel estimates and error variances, respectively, but for a model with state dimension $d = 32$. Figure 7.9(c)

[8] There are 260 nonlocal linear functionals in this problem, 4 representing nonlocal measurements and 256 representing the the 4×4 coarse-scale pixels tiling the 64×64 field.

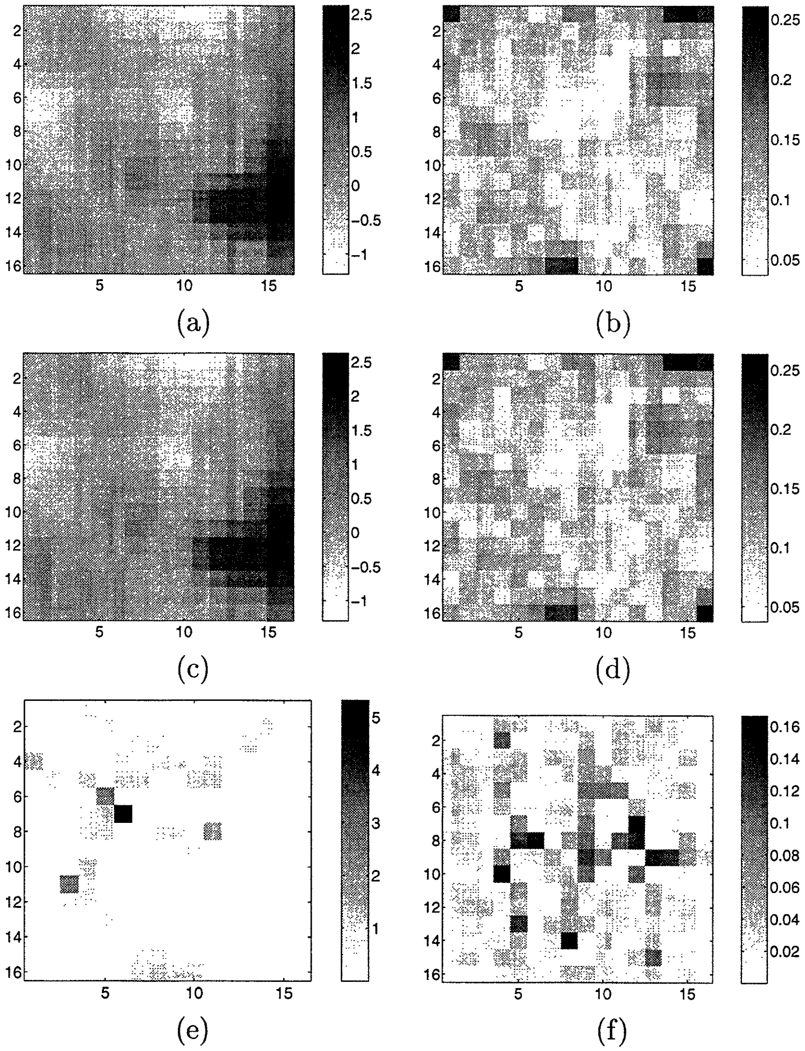


Figure 7.9. (a) Coarse-scale estimates (4 × 4 averages) based on the measurement geometry of Figure 7.7(b) ($d = 64$). (b) Error variances associated with the estimates of (a). (c) and (d) represent processing similar to (a) and (b) but with a MAR model with state dimension $d = 32$. (e) Absolute value of the fractional difference between (a) and (c). (f) Absolute value of the fractional difference between (b) and (d).

appears to be very similar to Figure 7.9(a), and Figure 7.9(b) appears to be very similar to and Figure 7.9(d). However, there are some significant differences. These can be seen in Figure 7.9(e) and Figure 7.9(f). The former illustrates the absolute value of the fractional difference between Figure 7.9(a) and Figure 7.9(c) where Figure 7.9(a) is used for normalization. That is, if F_a and F_c represent Figure 7.9(a) and Figure 7.9(c), respectively, then

$$\left| \frac{F_a - F_c}{F_a} \right| \tag{7.27}$$

is plotted in Figure 7.9(e) where the subtraction, division, and absolute value in (7.27) is pixel-wise. Similarly, Figure 7.9(f) illustrates the absolute value of the fractional difference between Figure 7.9(b) and Figure 7.9(d) where Figure 7.9(b) is used for normalization. Figure 7.9(e) and Figure 7.9(f) show that for many pixels the normalized difference between the 64-th order model (of Figure 7.9(a) and Figure 7.9(b)) and the 32-nd order model (of Figure 7.9(c) and Figure 7.9(d)) is insignificant. However, for some pixels the normalized difference *is* significant, indicating that some important statistical features are not captured by the lower-dimensional model.

The examples of this section have shown that, in certain circumstances, one can achieve excellent estimation results with lower dimensional models (e.g., using the GDM technique of Section 7.3) or with models for which nonlocal functionals are only incorporated approximately (e.g., using the AN technique of Section 7.1). In some cases, exact incorporation of nonlocal functionals is required (e.g., using the EN technique of Section 7.2) to achieve good performance. Further development of these techniques and characterizing the circumstances in which they are appropriate is an open problem.

Chapter 8

Contributions and Suggestions

THIS thesis makes significant contributions in the general area of stochastic modeling and, in particular, to the theory of internal MAR processes and MAR model identification. These are summarized in Section 8.1. Notwithstanding these contributions, there are a number of important open problems associated with the MAR framework. Some of these are extensions of work presented in this thesis, and these too are summarized in Section 8.1. Other suggestions for future research are found in Section 8.2.

■ 8.1 Thesis Contributions and Extensions

This thesis addressed three main topics in stochastic modeling with MAR processes. These topics were summarized in Section 1.2 as:

- computationally efficient internal MAR stochastic realization (Chapter 3, Chapter 4, Chapter 7),
- unification of the wavelet and MAR frameworks (Chapter 5), and
- covariance extension (Chapter 6).

Each of these main topics can be further subdivided into several problems, the solutions of which represent important contributions in their own right. Indeed, the resulting algorithms will serve as invaluable tools to researchers wishing to apply the MAR framework. Additionally, the conceptual framework developed in this thesis for addressing these problems represents a fertile foundation for future theory and algorithms. In this section we summarize these problems, their solutions, and possible extensions.

A Theory for Internal MAR Processes

Non-internal (i.e., external) MAR models have been developed and successfully applied to several signal processing problems. For example, the $1/f$-like models described in [28,29] and applied to one-dimensional [28,29] and two-dimensional [127,128] optical flow, oceanographic problems [63,66], and surface reconstruction [63,69] are external.

Additionally, high-fidelity and low-dimensional external models for fBm and other processes are developed in [37, 40, 96, 98, 100]. These facts beg the question: why restrict attention to internal models?

There are several important reasons to focus on the property of internality, many of which were discussed in Section 1.2.1 and Chapter 3. Included among these are the following:

- internality vastly simplifies model identification and reduces the problem to one of finding internal matrices;
- internality provides model consistency which is of critical importance when designing states to include specific nonlocal linear functionals (e.g., wavelet-based functionals, tomographic (line-integral) functionals, area averages, etc.); and
- internality leads to a scale-recursive notion of Markovianity and thereby plays a role in achieving computationally efficient realization algorithms.

In addition to these reasons there are others. One is that, given noisy (local or nonlocal) observations of the fine-scale process x^M, only the internal part of a state may be estimated. That is, any external component is in the null-space of the estimator. Another and more fundamental reason to focus on internality is that it is an important pillar in the foundation of state-space stochastic realization theory. Indeed, much of the success that state-space modeling has enjoyed stems from theoretical and algorithmic simplifications that internality provides. This thesis has shown that a MAR stochastic realization theory based on the concept of internality possesses many of the attractive features of its state-space counterpart. It is fair to say, then, that internality is an important concept in its own right and one that deserves precise characterization and development.

Chapter 3 provided such a development. The main result is that the correct parameterization of internal MAR states is through local internal matrices that linearly relate each state to the sub-process at the previous finer scale. This new parameterization differs from previous attempts to parameterize internal MAR states [96, 98, 100] in which each state is parameterized by a matrix that relates it to the finest scale subprocess. Not only did these previous approaches lead to inconsistent (i.e., external) models, they lead to computationally burdensome realization algorithms because the determination of the linear functionals relating each state to the fine scale relies on a number of variables proportional to the number of fine-scale nodes. In contrast, the scale-recursive parameterization developed in Chapter 3 leads to an efficient and consistent realization algorithm based on a scale-recursive notion of Markovianity (and on predictive efficiency).

Notwithstanding the motivation for and the success of the theory of internal MAR processes developed in this thesis, a corresponding theory for external MAR processes is attractive. Perhaps the most compelling motivation for the development of such a theory is the fact that to find a minimal MAR model (one with smallest possible state dimensions) one may need to look beyond internal models and consider the larger class of external ones [96]. However, as of this writing, there is no known path through the wilderness of external processes and, therefore, there is no known systematic procedure for searching for minimal models. The external models mentioned at the start of this section were all developed either by ad hoc means or as an unintended consequence of an improper parameterization for internal processes.

One possible (but, as of yet, not well thought out) approach to understanding external processes stems from the following observation. An external state can be written as a sum of an internal part and a (purely) external part. Therefore, beginning with a state of an exact, internal model, it is, perhaps, possible to add external randomness to achieve a lower dimensional state that fulfills the same decorrelation role as the original internal one.

Consistent and Efficient MAR Model Realization

To harness the utility of the MAR signal processing algorithms, one must translate the problem at hand into the MAR framework. Doing so means endowing a MAR process with all (or at least the most salient) statistical features of the underlying processes. For problems which can be characterized exclusively by the second-order statistics of a fine-scale random process (i.e., there are no nonlocal variables of interest), this corresponds to choosing parameters of a MAR process so that its fine-scale variables have second-order statistics that well-approximate the given ones.

It is precisely this MAR stochastic realization problem that is the topic of [40,96,98, 100]. However, the approach taken in these works suffers from several weaknesses, two of which are addressed by the algorithm developed in Chapter 4 of this thesis: consistency and computational complexity. A general-purpose algorithm based on the notion of fine-scale Markovianity was presented in Chapter 4 with complexity quadratic in problem size. Application of the boundary approximation further reduced this complexity to linear in problem size. In contrast, the general-purpose algorithm of [96, 98, 100] has complexity quartic in problem size. Another feature of the approach of Chapter 4 not shared by others is that it leads to internal (and, hence, consistent) models.

One important component of the model identification approach of Chapter 4 is predictive efficiency (Section 2.4). This estimation-theoretic tool is used to conditionally decorrelate random vectors and also plays a role in computational efficiency. The use of predictive efficiency in MAR model identification leads to a computational simplification of two orders as compared to using canonical correlations (the method used to conditionally decorrelate random vectors in previous MAR model identification approaches) because it treats the two vectors to be decorrelated asymmetrically. In the context of MAR stochastic realization, this asymmetry permits the organization of calculations so that the inversion and singular value decomposition of large matrices is not required—steps that cannot be avoided when using canonical correlations.

Additional computational savings are achieved by the boundary approximation in which one only attempts to conditionally decorrelate elements of a random process from nearby elements, rather than from all other elements. The intuition that motivated the boundary approximation is that for nearly Markov processes or processes with quickly-decaying long-range correlations the boundary elements summarize the salient statistical features. In Section 4.2.2 some theoretical justification for this intuition was provided in the form of bounds on the approximation error. While the bounds do show that as the process becomes closer to Markov (in a certain sense) or has faster decaying long-range correlations the boundary approximation is closer to the exact method, they are rather crude and certainly far from tight. Therefore, one obvious extension is to refine and tighten these bounds. Another extension stems from the observation made in Section 4.3 that the boundary approximation is an accurate one for a richer class of processes than Markov or quickly decorrelating ones. Characterizing the class of processes for which the boundary approximation is accurately applicable is an open problem.

The boundary approximation is not the only kind of approximation one can imagine making. Indeed, taking a broader view of the problem suggests some alternatives. The problem is to estimate a large vector z_2 from a small one z_1 and, therefore, the estimate lives in a subspace with dimension equal to the length of z_1. Hence, it is reasonable to expect that consideration of the estimate of a low-dimensional summary of z_2 ought to be sufficient. There are a variety of ways, other than the boundary approximation, to reduce the dimensionality of z_2 and the heart of the problem is to find ones that are both useful and efficient. One idea is to use Krylov subspace methods [84,161,163,164] to project z_2 onto a small subspace that captures most of the variability that can be estimated from z_1. Another idea is motivated by multipole-like or mean-field-like approximations [64,65,87,196–198]. Approximations of this type summarize distant variables by aggregating them rather than ignoring them as the boundary approximation does. Therefore, while they provide a means to account for some of the statistical features of distant variables, they are not as greedy or as computationally costly as exact methods or as myopic as local methods such as the boundary approximation. Developing efficient and effective alternatives to the boundary approximation and characterizing the class of processes for which they make sense is an open problem.

Unification of Wavelet and MAR Frameworks

The unification of wavelets and MAR processes provided in Chapter 5 is both theoretically satisfying and practically important. It is satisfying because, while wavelets motivated the development of MAR processes and while the two frameworks are tantalizingly similar, they seemed, until this thesis, irreconcilable except in the case of the Haar wavelet. It is important because MAR-wavelet processes combine the renowned modeling power of wavelets with the flexibility and efficiency of the MAR framework. Additionally, MAR-wavelet processes provide another route toward stochastic modeling with the MAR framework—a route that is less closely tied to the minute and (in many cases) irrelevant or poorly known details of the statistics of the signal being modeled.

There were two issues to address to achieve the unification. The first was model consistency (i.e., internality) which is crucial if MAR state elements are to be wavelet and scaling coefficients of the fine-scale process. Once internality was understood (Chapter 3), the second issue was how to design the MAR-wavelet states associated with any compactly supported orthogonal or biorthogonal wavelet to provide internality while keeping the state dimension low. This latter issue is resolved in Section 5.2.2.

In Section 5.3 MAR-wavelet processes were applied to the problem of stochastic realization. One main result is that internal MAR-wavelet models (for which the MAR process driving noise represents the prediction error in estimating detail coefficient from coarser scale scaling and detail coefficients) outperform standard MAR-wavelet models (for which the MAR process noise represents the detail coefficients themselves). This result can be explained as follows. While the standard MAR-wavelet models make the assumption that wavelets perform a Karhunen-Loeve decomposition of the process being modeled, the internal MAR-wavelet models are more sophisticated. Indeed, they make the weaker assumption that the parent-to-child *prediction errors* are white and thereby incorporate a synthesis algorithm for the detail coefficients complementing the customary one for the scaling coefficients. An important consequence of the fact that internal MAR-wavelet models exploit the correlations among neighboring wavelet coefficients is that models of relatively low dimension, corresponding to wavelets of small support and few vanishing moments, can achieve a surprisingly high degree of statistical fidelity.

The MAR-wavelet processes developed in Chapter 5 are applicable to the modeling of one-dimensional signals. The extension to two dimensions poses a number of important and challenging problems, some of which are currently being studied [42]. Among these are the two-dimensional generalizations of Proposition 5.2.1 and Proposition 5.2.2. Even a casual glance at the proof of Proposition 5.2.1 in Appendix C makes clear that this generalization is far from straightforward.

The MAR-wavelet work of Chapter 5 is an instance of a more general concept, namely the idea of selecting internal matrices from a library of linear functionals. This idea is an extremely important one in many applications, some of which have been highlighted in this thesis. For example, if one is interested in estimating nonlocal variables from data collected at multiple resolutions, the MAR states must be able to represent the coarser variables in addition to playing a conditional decorrelation role. This corresponds to selecting internal matrices from a library of linear functionals which include ones that not only do a good conditional decorrelation job but also include ones that represent nonlocal variables to estimate or to include as measurements. The problem of developing and selecting from a more general (not necessary wavelet-based) library suitable for a rich class of modeling problems is an open one.

Model Identification from Incomplete Covariance Information

All previous approaches to MAR stochastic realization rely on the assumption that the covariance matrix of the signal to be modeled is precisely and completely known. Relaxation of this assumption is critical if the MAR framework is to be applied to large real-world problems—ones for which full specification of a covariance matrix is unlikely to be available. Consideration of the problem for which only a subset of the covariance elements are known leads to the covariance completion (or extension) problem which was the focus of Chapter 6.

To address the specific problem of building a MAR model for the maximum-entropy completion of a partially specified (banded) covariance matrix, we developed a generalization of the classical Levinson algorithm in Section 6.4. In doing so, we showed that completions of *any* partially specified covariance matrix for which the known entries correspond to a chordal graph may be computed via one-element extensions. Applying this generalized-Levinson algorithm to the MAR maximum-entropy completion modeling problem, we showed that the parameters of a MAR model can be computed efficiently. Indeed, the complexity of doing so is comparable to that which is achieved by the classical Levinson algorithm in computing the parameters of an autoregressive model.

The specific problem considered in Chapter 6 represents a first step in the development of techniques for MAR model identification from incomplete covariance information. Indeed, there are many conceivable extensions and open problems. The most obvious one, perhaps, is the relaxation of the rather rigid requirement of Chapter 6 that the $2k + 1$ main diagonal bands are known. The banded partial covariance case is a convenient one because its maximum-entropy completion corresponds to a Markov process, and the internal matrices that define MAR states are clear. One challenge in considering more a more general (but still chordal) pattern of known covariance elements is that, in general, it is unclear what the internal matrices ought to be. The same problem arises if one considers completions other than the maximum-entropy one.

A seemingly more difficult challenge arises when one considers covariance information that *does not* correspond to a chordal graph. This situation may occur because, in addition to information about the correlation between fine-scale random variables, the correlation between some coarse-scale, nonlocal variables are also provided. Non-chordal graphs also arises naturally in two-dimensions where nearest-neighbor covariance information does not give rise to a chordal graph (recall the nearest-neighbor graph of Figure 6.2(c)). However, non-chordal graphs can arise in important one-dimensional problems as well, for instance in two-point boundary value problems [5] such as the Brownian bridge which has applications in finance [176], polymer science [136], thermodynamics [150], and immunization theory [15]. In the Brownian bridge problem, in addition to diagonal bands, the covariance between the end-points of a one-dimensional random process are known.

At first glance, the Brownian bridge problem seems like a simple variation on the one considered in Chapter 6 since it involves only one additional covariance element—the one associated with the end-points of the process. Heightening this sense of simplicity, the process end-points are both available in the root node state of the MAR end-point model used in Chapter 6. Therefore, it would seem that knowledge of the additional covariance element only impacts the computation of the root node statistics. Unfortunately, however, once we leave the world of chordal graphs the *entire* procedure for computing the extension necessary to build the MAR model must be revamped. The reason is that *all* of the theory of Chapter 6 is premised on chordality and it is unclear how to modify it or the resulting extension algorithm for the case of non-chordal graphs. Some optimism can be gleaned from the fact that there has been some work on non-chordal completions [10, 33, 82, 159]. Other covariance completion work different from that previously discussed in this thesis is found in [2] and [132]. The former considers the completion of banded partial covariance matrices for which there are gaps (i.e., non-adjacent bands are known). The latter considers the completion of partial covariance matrices under linear constraints on the unknown elements. Harnessing these techniques for application to MAR model identification is an open problem.

There are a number of open graph-theoretic questions raised by the covariance extension work of Chapter 6. One is the development of fast algorithms for finding chordal sequences. Another is the existence and construction of efficient chordal sequences (which are defined by Definition 6.4.1). While solutions to these problems are not necessary for the particular problem addressed in Section 6.5 (that of building a MAR model for the maximum-entropy completion of a banded covariance matrix), they are important for the efficiency of the generalized-Levinson algorithm developed in Section 6.4.

Another type of problem arises when considering the identification of MAR models from *no* covariance information, i.e., when considering identification from directly from data. No doubt many of the techniques developed in this thesis, including those of covariance extension, have data-based counterparts. Additionally, there are likely approaches not found in this thesis that are better suited to model identification from data—for instance, approaches based on iterative learning algorithms or adaptive filtering which are currently being investigated [180]. Although this work is preliminary, it appears that one of the fundamental difficulties is finding internal matrices. One possible way around this difficulty is to simply select internal matrices from a class that is known to be broadly effective (e.g., the class of wavelet bases or the class given by end-point MAR models for Markov processes). Then these initially selected internal matrices can, perhaps, be evolved iteratively in a data-driven way to arrive at ones more appropriate for the particular statistics of the data.

Incorporation of Nonlocal Variables

The challenge of MAR model identification resides in the number of conflicting constraints that a MAR model must satisfy. These include: possessing low dimensional states, achieving high statistical fidelity (equivalently, providing a good approximation to Markovianity), and possessing internality (or, consistency). Consideration of these constraints is sufficient to address the modeling of a fine-scale random process. However, to address estimation problems involving nonlocal variables one must add another constraint: that each of the specific nonlocal variables of the problem at hand be expressible (at least to good approximation) as a linear function of a MAR state. Dealing with this additional constraint is the topic of Chapter 7.

Three techniques are developed in Chapter 7. The approximate nonlocal (AN) method of Section 7.1 incorporates nonlocal variables approximately and results in no increase in state dimension over that which is required to model fine-scale statistics. As is shown by example in Section 7.4, there are problems for which approximate representation of nonlocal measurements has an acceptably small impact on the accuracy of the estimates and error variances. Approximating measurements is also consistent with the point of view that *any* model is an idealization of the true statistics. Therefore, this provides motivation and justification for making approximations for the sake of computational efficiency if the approximation remains faithful to the statistical features that are important for the problem at hand. A precise characterization of when such approximations are effective is an open issue.

Another technique developed in Chapter 7, the exact nonlocal (EN) method of Section 7.2, incorporates nonlocal variables exactly, a problem that has been considered in [37,39]. One significant difference between the EN method and that of [37,39] is that the latter begins with a model while in the former a model is built around the nonlocal linear functionals. Therefore, the information carried by the nonlocal linear functionals may be conditioned upon leading to a more efficient allocation of state dimension and higher fidelity models.

The final technique of Chapter 7, the goal directed modeling (GDM) technique of Section 7.3, exploits the structure of the estimation problem to simplify model realization. We showed with an example that there are instances in which the Markov property leads to a poor allocation of resources because it conditionally decorrelates random variables that are irrelevant for estimation. GDM is based on an intellectual successor to the Markov property and our example showed that a model designed using this technique can outperform one based on the Markov property. A major unresolved issue associated with this technique is computational complexity. The challenge in assessing complexity is to understand the impact of the number, type, and node placement of nonlocal linear functionals. While the modeler is free to select the nodes at which to represent nonlocal variables, it is unclear how to make this choice to minimize the complexity of model realization or of estimation.

The work of Chapter 7 suggests a modeling philosophy substantially different from those of previous chapters—that of designing a model with a specific purpose in mind (e.g., an estimation problem or a class of related estimation problems). In turn, this philosophy suggests fundamental questions. It is clear from the work of Chapter 7 that, given the additional structure provided by a particular modeling goal, one can achieve models better suited for that goal as compared to those designed in the absence of this additional structure. However, it is far from clear precisely how one ought to exploit this additional structure or even what all the degrees of freedom are. The concept of goal directed modeling presents significant problems many of which are raised but not conclusively resolved by the work in Chapter 7.

■ 8.2 Additional Suggestions for Future Research

While many extensions of the work presented in this thesis were discussed in the previous sections of this chapter, there are a number of other open issues associated with the MAR framework. Some of these are discussed in this section.

Variations of and Alternatives to Predictive Efficiency

In Chapter 4 we based our approach to conditionally decorrelating several random vectors on predictive efficiency. The only aspects of predictive efficiency that are crucial to stochastic realization are computational efficiency and an interpretation as a measure of conditional decorrelation or distance from Markovianity. It is conceivable that there are other criteria (e.g., those based on information-theoretic concepts like mutual information) that also have these properties and which may lead to more accurate (in some sense) realizations or are more appropriate for other types of MAR model identification problems not discussed in this thesis (like the estimation of MAR parameters from data).

In the remainder of this section we discuss some open problems associated with predictive efficiency. As stated in Section 2.4, finding a procedure for solving the higher-order predictive efficiency problem (2.64) is an open problem. Our suboptimal solution of solving several pair-wise problems also raises issues. One issue is how to choose r_i, the number of linear functionals of z_i to keep. Our approach of choosing r_i implicitly by keeping the linear functionals corresponding to the d highest eigenvalues has one unfortunate consequence: the collection of linear functionals may contain redundant information.

A way to avoid this redundancy is to consider adding linear functionals sequentially where at each sequential step we add one or more linear functionals that have the highest predictive efficiency conditioned on the linear functionals that have been chosen during previous steps. One simple way to do this is first to incorporate linear functionals from z_1, then from z_2, etc., an approach that requires specifying the r_i sequentially rather than collectively. This will produce models that depend on the order in which the z_i appear in the sequence. More complex alternatives (e.g., cycling through the z_i several times, incorporating smaller numbers of linear functionals at each step) can potentially achieve greater statistical fidelity with an increase in computational load.

Another way to determine the r_i is minimize the maximum cost of the pair-wise problems. Since the predictive efficiency matrices (U_i, Λ_i) for the pair-wise problems (2.65) do not depend on one another, we may compute them all before determining the $\{r_i\}$. Then, letting λ^i_j be the j-th eigen-value in Λ_i, consider the discrete optimization problem

$$\min_{\{r_i\}} \max_{i\in[1:q]} \bar{\varepsilon}\big(z_i^c \mid V_{i,r_i} z_i\big) = \min_{\{r_i\}} \max_{i\in[1:q]} \sum_{j=r_i+1}^{d} \lambda^i_j \tag{8.1}$$

subject to the constraint that $\sum_{i=1}^{q} r_i \leq d$. It is worth emphasizing that there is no guarantee that the matrix

$$\bar{V} \triangleq \operatorname{diag}(V_{1,r_1}, V_{2,r_2}, \dots, V_{q,r_q}) \tag{8.2}$$

which is formed based on these "optimal" $\{r_i\}$ is optimal in the sense of minimizing the cost function of (2.64) which we repeat:

$$\bar{\varepsilon}\big(z_1, z_2, \dots, z_{q+1} \mid \bar{V}z_0\big)\,. \tag{8.3}$$

There may be another choice for the $\{r_i\}$ that both decreases (8.3) yet increases the cost function of (8.1). How to select the $\{r_i\}$ to minimize (8.3) is an open problem.

Overlapping with Nonlocal Variables

We reviewed the overlapping tree method of [63, 96, 97] in Section 2.3.4 and applied it to reduce the blockiness of our examples in Chapter 4. Although not employed in this thesis, the overlapping method can be applied to estimation, not just sample-path generation. However, a significant limitation of the current understanding of estimation with overlapping is that it is unclear how to apply the method to problems with nonlocal measurements—problems of the type discussed in Chapter 7. That is, to date the overlapping method has only been applied to fine-scale estimation problems based on point-wise measurements.

To see the difficulty in extending the overlapping method to the case of nonlocal measurements we first review how the method is applied in the case of point-wise measurements. For this purpose, consider Figure 2.6(b). This figure depicts an overlapping tree with four fine-scale nodes, however the underlying fine-scale process $f^M = \big[f^M(1) \quad f^M(2) \quad f^M(3)\big]^T$ is length-three. The sample $f^M(2)$ is mapped to two different fine-scale nodes, s and t. Suppose we wish to use a MAR model indexed by the nodes of this tree for an estimation problem involving a point-wise measurement of $f^M(2)$. The way this is handled in [63, 96, 97] is to duplicate this measurement and incorporate it at *both* nodes s and t. Duplicating the measurement in this way seems to imply that twice the amount of information will be provided to the estimator. However, this is easily remedied by doubling the variance of the measurement noise.

Now consider an estimation problem involving the nonlocal measurement

$$g = f^M(1) + f^M(2) + \nu \tag{8.4}$$

where ν is measurement noise. It is clear from Figure 2.6(b) that a natural node at which to represent g is $s\bar{\gamma}$ since both $f^M(1)$ and $f^M(2)$ are represented by states residing at nodes descending from $s\bar{\gamma}$. However, $f^M(2)$ is also represented by a state indexed by a node (namely t) that descendents from $t\bar{\gamma}$. Therefore, in some sense, part of the measurement g should be represented at $t\bar{\gamma}$. Additionally, the measurement noise also ought to be somehow divided up among $s\bar{\gamma}$ and $t\bar{\gamma}$ in such a way so as not to increase the overall amount of information provided to the estimator.

Based on the current theory of overlapping trees, it is unclear, in general, how to distribute nonlocal measurements (or parts thereof) among coarse-scale nodes in an overlapping tree. However, for the particular problem described in the previous paragraph there is a simple solution. The measurement g can simply be represented at the root node since all of states representing $f^M(1)$ and $f^M(2)$ descend from the root node. This suggests a way to handle more general problems involving nonlocal measurements. For any overlapping tree and any nonlocal measurement $g = h^T f^M + \nu$ there is a unique node s with maximal scale such that the descendents of s index *all* of the variables in the tree that comprise $h^T f^M$. Therefore g may be unambiguously represented at node s or at any ancestor of node s. Evaluation of the merit of this idea, fleshing out the theoretical details, and turning this idea into an efficient algorithm is left for the future.

Global Error

The MAR stochastic realization techniques developed in this thesis, as well as those previously developed, myopically focus on *local* criteria when designing MAR states. However, typical measures of a MAR model's fidelity and utility are global (e.g., measures such as the complexity of the MAR estimator, the element-wise maximum of $|P_{f^M} - P_{x^M}|$, or mean-square estimation error). Since currently available realization algorithms do not specifically minimize a global error criterion, when state reduction is done, there is no clear way to tell how the reduction will affect the overall degree of approximation. In this section we present some preliminary analysis which may be of use in future work on developing a realization approach to minimize a global error criterion.

One natural and less myopic, but as of yet elusive, realization approach is to choose internal matrices while minimizing the complexity, c, of the estimation algorithm which is the sum of the cubes of the state dimensions:

$$c = \sum_{s \in \mathcal{S}_0} d_s^3 . \tag{8.5}$$

Since, in some cases, it is important to obtain a realized fine-scale covariance that is close to P_{f^M}, another approach is to constrain the degree of approximation of the realization. That is, to minimize

$$\varepsilon_{\mathrm{g}} = \|P_{f^M} - P_{x^M}\| \tag{8.6}$$

where $\|\cdot\|$ is an appropriate norm. The quantity ε_{g} is referred to as the *global error.* Combining the two criteria (minimizing c subject to a constraint on ε_{g} or vice versa) is a reasonable approach to take. We emphasize that this approach is decidedly *not* the right way to proceed in the context of certain estimation problems. As discussed in Chapter 7, in the context of some estimation problems, designing a MAR model to match a specific fine-scale covariance may put resources (modeling fidelity) where it is not needed. In such circumstances, a notion of global error that is related to estimation error is more relevant. However, for the not insignificant class of problems where it *is* important to match a pre-specified fine-scale covariance (e.g., for the purpose of sample path generation or because the data and estimates are dense and fine-scale) minimizing some measure of global error ε_g and/or controlling the overall complexity c makes sense.

While this global error problem is far from solved, there is a scale-recursive decomposition of $P_{f^M} - P_{x^M}$ which may present a way to break the problem down into more manageable pieces. To present this scale-recursive decomposition, we need a few definitions. Using the MAR dynamics we can write

$$x^n = \mathcal{A}_n x^{n-1} + w^n \tag{8.7}$$

where, as always, x^n is the process at scale n. Also, $\mathcal{A}_n$ is a block matrix whose blocks consist of all the $A(s)$ matrices for $m(s) = n$. The noise w^n has a block diagonal covariance matrix $\mathcal{Q}_n = \text{diag}\big(Q(s_0), Q(s_1), \ldots, Q(s_{q^n-1})\big)$ where $\imath(s_k) = k$, the shift of node s_k and $m(s_k) = n$ for all k. Using (8.7), we define the *scale transition matrix*, $\Upsilon(\cdot,\cdot)$ as

$$\Upsilon(i,j) \triangleq \begin{cases} I & \text{if } i = j\,, \\ \mathcal{A}_i \Upsilon(i-1,j) = \Upsilon(i,j+1)\mathcal{A}_{j+1} & \text{if } i \geq j\,, \\ \Upsilon(j,i)^T & \text{if } i \leq j\,. \end{cases} \tag{8.8}$$

Next, we define P_n^m to be the realized covariance at scale n based on the covariance, P_{f^m} (defined in Chapter 4) at scale $m \leq n$. What we mean by this is that, if we begin with the P_{f^m} as the covariance for scale m and use the dynamics to propagate from scale m to n, the resulting covariance at scale n is P_n^m. Therefore, the realized covariance at scale n, P_{x^n}, can be written equivalently as P_n^0. It follows from (8.7) that

$$P_n^m = \Upsilon(n,m) P_{f^m} \Upsilon(m,n) + \sum_{k=m+1}^{n} \Upsilon(n,k)\mathcal{Q}_k \Upsilon(k,n)\,. \tag{8.9}$$

The error associated with P_n^m is defined as $\widetilde{P}_n^m$ where

$$\widetilde{P}_n^m \triangleq P_{f^n} - P_n^m\,. \tag{8.10}$$

The matrix $\widetilde{P}_n^m$ represents the accumulation of errors induced by each scale between n and m. We can extract the contribution to this error from one scale by taking the difference $\widetilde{P}_n^m - \widetilde{P}_n^{m+1}$. We define this difference to be $\check{P}_n^m$. Some algebraic manipulation reveals that

$$\check{P}_n^m \triangleq \widetilde{P}_n^m - \widetilde{P}_n^{m+1} \tag{8.11a}$$

$$= \Upsilon(n,m+1)\left[P_{m+1} - \mathcal{A}_{m+1} P_m \mathcal{A}_{m+1}^T - \mathcal{Q}_{m+1}\right]\Upsilon(m+1,n)\,. \tag{8.11b}$$

Intuitively this makes sense since term in the brackets is $\widetilde{P}^m_{m+1}$ and this is projected down to scale n with the scale transition matrix.

Finally, by definition of $\check{P}^m_n$, we can decompose the error at any scale as a sum of errors contributed by coarser scales:

$$\widetilde{P}^m_n = \sum_{k=m}^{n-1} \check{P}^k_n \tag{8.12}$$

where we have used the fact that $\widetilde{P}^n_n = P_{f^n} - P^n_n \equiv 0$. In particular we can expand this out for the case where $n = M$ and $m = 0$:

$$\widetilde{P}^0_M = \sum_{k=0}^{M-1} \check{P}^k_M \tag{8.13a}$$

$$= \sum_{k=0}^{M-1} \Upsilon(M, k+1) \left[P_{k+1} - \mathcal{A}_{k+1} P_k \mathcal{A}^T_{k+1} - \mathcal{Q}_{k+1}\right] \Upsilon(k+1, M)\,. \tag{8.13b}$$

Since the global error is $\varepsilon_{\mathrm{g}} = \|\widetilde{P}^0_M\|$, one possible way to build in some control over global error is to attempt to minimize some norm of the scale-to-scale errors $\widetilde{P}^k_{k+1} = P_{k+1} - \mathcal{A}_{k+1} P_k \mathcal{A}^T_{k+1} - \mathcal{Q}_{k+1}$. The computational complexity of computing all of these scale-to-scale errors is $O(N^2)$. To see this, note that $\widetilde{P}^k_{k+1}$ has no more than $d^2 q^{2k+2}$ elements and that $\mathcal{A}_{k+1}$ has no more than d non-zeros per row. Thus we can compute $\widetilde{P}^k_{k+1}$ with $O(d^5 q^{2k+2})$ operations. Summing over all scales yields

$$d^5 q^2 \sum_{k=0}^{M-1} (q^2)^k = O(d^5 q^{2M})\,. \tag{8.14}$$

Since the number of elements in the fine-scale process is $N = dq^M$ this translates to $O(d^3 N^2)$ complexity. If d is chosen independent of N the complexity is $O(N^2)$.

Precisely how to use the scale-to-scale errors is an open research problem. One possibility which has yet to be explored is to use the scale-to-scale errors in an iterative procedure to allocate state dimensions. The idea is to build a model with some nominal state dimensions and then to compare (somehow) the scale-to-scale errors. The states indexed by nodes of scales which give rise to relatively larger errors can be increased while states indexed by nodes of scales which contribute relatively smaller errors can be decreased. This procedure may be iterated until the scale-to-scale errors are all of the same magnitude.

An exploration of the type described in the previous paragraph could yield deep results about how state dimensions ought to scale with problem size. Specifically, given a particular class of processes (fBm, say) and an error budget, one may be able to specify what the state dimensions ought to be as a function of problem size in order to meet the error budget. Conversely, given an overall complexity constraint (the sum of the cubes of the state dimensions, say), one may be able to say something about how to allocate state dimensions in order to minimize global error.

Characterizing the Limitations of the MAR Framework

The theory and algorithms developed in this thesis have substantially broadened the range of problems to which the MAR framework may be applied. This breadth notwithstanding, the MAR framework cannot efficiently address *every* problem. Clearly the purview of the MAR framework does not extend beyond the class of linear problems (or those which can be effectively linearized) that can be well-characterized by second-order statistics. However, even within the class of second-order/linear problems there are certainly some that are not well-suited to the MAR framework because, perhaps, they require impractically large state dimensions. Characterizing the limitations of the MAR framework is a challenging and important open problem.

Pairing this challenge down may provide some degree of simplification. For instance, consider the problem of MAR model realization for a fine-scale process (as opposed to model realization with the additional constraint of incorporating nonlocal coarse-scale variables). MAR models offer substantial computational advantages for statistical inference as long as they have state dimension that is small relative to the problem size, N. This raises the question: what class of fine-scale processes can be captured with a MAR model with state dimension that is independent of (or a slowly growing function of) N?

Perhaps one approach to addressing this problem is to restrict attention to a specific parameterized class of MAR models for which the internal matrices are already defined (e.g., the MAR-wavelet models of Chapter 5). Doing so permits one to rephrase the problem as: what processes can be well-modeled (i.e., realized at the fine scale) using models chosen from this parameterized class? For the specific class of MAR-wavelet processes one (incomplete) answer to this question is: all processes whose Karhunen-Loeve transform is well-approximated by the wavelet transform can be well-modeled. However the class of processes well-modeled by MAR-wavelet models is, no doubt, substantially larger than this.

Questions of the type posed in this section are, in some sense, the deepest open ones concerning the MAR framework and the work presented in this thesis represents one step toward their resolution.

Appendix A

Proofs for Chapter 3

In this appendix a proof of Proposition 3.1.2 (Section A.1) is provided as well as the completion of the proof of Proposition 3.2.2 (Section A.2).

■ A.1 Proof of Proposition 3.1.2

Proof. We begin with the "only if" direction. Given that $x(\cdot)$ is a locally internal MAR process, it has dynamics of the form (2.14). Thus, for all $s \in \mathcal{S}_0 - \mathcal{T}_0(M)$

$$x(s\alpha_i) = A(s\alpha_i)x(s) + w(s\alpha_i)\,. \tag{A.1}$$

Since $w(\cdot)$ is white and uncorrelated with $x(0)$, it follows that $w(s\alpha_i)$ is uncorrelated with $x(s)$. Therefore, (A.1) represents the linear least-squares estimate of $x(s\alpha_i)$ from $x(s)$ plus the estimation error $w(s\alpha_i)$. Then, (3.6) follows from (3.2) together with standard linear least-squares formulae.

To show the "if" direction, notice that (3.6) implies that by the MAR dynamics

$$x_s^{m(s)+1} = \underbrace{\begin{bmatrix} J_{s\alpha_1} \\ J_{s\alpha_2} \\ \vdots \\ J_{s\alpha_q} \end{bmatrix}}_{I} P_{x_s^{m(s)+1}} V_s^T (V_s P_{x_s^{m(s)+1}} V_s^T)^{-1} x(s) + \underbrace{\begin{bmatrix} w(s\alpha_1) \\ w(s\alpha_2) \\ \vdots \\ w(s\alpha_q) \end{bmatrix}}_{\triangleq w}\,. \tag{A.2}$$

Pre-multiplying (A.2) by V_s results in $V_s x_s^{m(s)+1} = x(s) + V_s w$. To conclude the proof we now show that the term $V_s w$ is zero. For notational simplicity, let us define

$$R \triangleq P_{x_s^{m(s)+1}} - P_{x_s^{m(s)+1}} V_s^T (V_s P_{x_s^{m(s)+1}} V_s^T)^{-1} V_s P_{x_s^{m(s)+1}}\,. \tag{A.3}$$

The covariance matrix for $V_s w$ is

$$\begin{aligned} \mathrm{E}[V_s w w^T V_s^T] &= V_s \operatorname{diag}\left[Q(s\alpha_1), Q(s\alpha_2), \ldots, Q(s\alpha_q)\right] V_s^T && \text{(A.4a)} \\ &= V_s \operatorname{diag}\left[J_{s\alpha_1} R J_{s\alpha_1}^T, J_{s\alpha_2} R J_{s\alpha_2}^T, \ldots, J_{s\alpha_q} R J_{s\alpha_q}^T\right] V_s^T && \text{(A.4b)} \\ &= V_s R V_s^T && \text{(A.4c)} \\ &= 0 && \text{(A.4d)} \end{aligned}$$

where the first equality follows from the definition of w in (A.2) and the second equality follows from the definition of $Q(s\alpha_i)$ given in (3.6) and of R. The third equality follows from the fact that R is block diagonal because it is the estimation error covariance matrix in estimating $x_s^{m(s)+1}$ from $x(s)$ and $x(s)$ conditionally decorrelates $\{x(s\alpha_i)\}_{i=1}^q$ by the Markov property. The fourth equality follows from the definition of R. Since $V_s w$ has zero-mean and zero covariance it is deterministically zero. This completes the proof. ■

A.2 Completion of Proof of Proposition 3.2.2

To complete the proof we need to show that for an arbitrary s in $\mathcal{S}_0 - \mathcal{T}_0(M)$, $x(s)$, which has the scale-recursive Markov property, conditionally decorrelates the vectors in the set $\{x^M_{s\alpha_i}\}_{i=1}^q \cup \{x^M_{s^c}\}$. We do this by induction starting at the next to finest scale (scale $M-1$) and proceeding to coarser scales. First, we note that the assertion is trivially true for $m(s) = M-1$ since the two sets $\{x(s\alpha_i)\}_{i=1}^q \cup \{x_{s^c}^{m(s)+1}\}$ and $\{x^M_{s\alpha_i}\}_{i=1}^q \cup \{x^M_{s^c}\}$ coincide. Next, suppose the assertion holds at scale n, that is, for all $s \in \mathcal{T}_0(n)$, $x(s)$ conditionally decorrelates $\{x^M_{s\alpha_i}\}_{i=1}^q \cup \{x^M_{s^c}\}$ and consider the case for which $s \in \mathcal{T}_0(n-1)$. For an arbitrary node s at scale $n-1$ and for an arbitrary child of s we have that

$$x^M_{s\alpha_i} = \widehat{\mathrm{E}}\left[x^M_{s\alpha_i} \mid x^n\right] + \widetilde{x}^M_{s\alpha_i} \tag{A.5a}$$

$$= \widehat{\mathrm{E}}\left[x^M_{s\alpha_i} \mid x(s\alpha_i)\right] + \widetilde{x}^M_{s\alpha_i} \tag{A.5b}$$

and

$$x^M_{s\alpha_i^c} = \widehat{\mathrm{E}}\left[x^M_{s\alpha_i^c} \mid x^n\right] + \widetilde{x}^M_{s\alpha_i^c} \tag{A.6a}$$

$$= \widehat{\mathrm{E}}\left[x^M_{s\alpha_i^c} \mid x^n_{s\alpha_i^c}\right] + \widetilde{x}^M_{s\alpha_i^c} \tag{A.6b}$$

where in these identities we've used the induction hypothesis. It follows that the errors $\widetilde{x}^M_{s\alpha_i}$ and $\widetilde{x}^M_{s\alpha_i^c}$ are uncorrelated with each other (due to the induction hypothesis) and with x^n (due to the orthogonality property of linear least-squares estimation). By assumption, $x(s)$ is an internal state, and it is a linear combination of its children. That is, we have that for some V_s, $x(s) = V_s x^n_s$.

We now use these facts to show that $x^M_{s\alpha_i}$ and $x^M_{s\alpha_i^c}$ are uncorrelated when conditioned on $x(s)$. By assumption, $x(s)$ conditionally decorrelates $x^n_{s\alpha_i^c}$ from $x(s\alpha_i)$. Therefore, referring to (A.5b) and (A.6b), the two terms $\widehat{\mathrm{E}}\left[x^M_{s\alpha_i} \mid x(s\alpha_i)\right]$ and $\widehat{\mathrm{E}}\left[x^M_{s\alpha_i^c} \mid x^n_{s\alpha_i^c}\right]$ are conditionally uncorrelated when conditioned on $x(s)$. As mentioned, the terms $\widetilde{x}^M_{s\alpha_i}$ and $\widetilde{x}^M_{s\alpha_i^c}$ are uncorrelated with each other and with x^n and therefore with $x(s) = V_s x^n_s$. Hence, it follows that $x^M_{s\alpha_i}$ and $x^M_{s\alpha_i^c}$ are uncorrelated when conditioned on $x(s)$. Since $s\alpha_i$ was an arbitrary child of s, this holds for all children and the proposition is proved.

Appendix B

Proofs for Chapter 4

In this appendix proofs of Proposition 4.1.1 (Section B.1), Proposition 4.2.1 (Section B.2) and Proposition 4.2.2 (Section B.3) are provided.

■ B.1 Proof of Proposition 4.1.1

Proof. Let Q be as defined in Lemma 4.1.1 and let R be the covariance matrix for $w = \begin{bmatrix} w(s\alpha_1)^T & w(s\alpha_2)^T & \cdots & w(s\alpha_q)^T \end{bmatrix}^T$ where $w(s\alpha_i)$ is as defined in (4.2). Then, since $x(\cdot)$ has the Markov property R has the form

$$R = \operatorname{diag}(Q(s\alpha_1), Q(s\alpha_2), \ldots, Q(s\alpha_q)) . \tag{B.1}$$

Note that R can be obtained from Q by setting all off-diagonal blocks to zero. (This, in fact, is precisely how the MAR model $x(\cdot)$ is obtained from the tree-indexed model $f(\cdot)$.) By an argument similar to the one made in the proof of Lemma 4.1.1 and also made in the proof of Proposition 3.1.2, $x(\cdot)$ is internal if and only if $V_s R V_s^T = 0$. We have from Lemma 4.1.1 that $V_s Q V_s^T = 0$. Notice that the m, n block of $V_s Q V_s^T$ is given by

$$[V_s Q V_s^T]_{m,n \text{ block}} = \begin{cases} V_{mm} Q(s\alpha_m, s\alpha_n) V_{nn}^T & \text{if } m \neq n\,, \\ V_{mm} Q(s\alpha_m) V_{mm}^T & \text{if } m = n \end{cases} \tag{B.2}$$

where we have used the fact that V_s is assumed to be block diagonal. It follows that $V_s R V_s^T = 0$. ■

■ B.2 Proof of Proposition 4.2.1

Proof.

$$\begin{aligned} \bar{\varepsilon}(J_k z_2 \mid V z_1) &= \operatorname{trace}(J_k P_{21} P_1^{-1} P_{21}^T J_k^T) - \operatorname{trace}\big(J_k P_{21} V^T (V P_1 V^T)^{-1} V P_{21}^T J_k^T\big) && \text{(B.3a)} \\ &\leq \operatorname{trace}\big(P_1^{-1/2} P_{21}^T J_k^T J_k P_{21} P_1^{-T/2}\big) && \text{(B.3b)} \\ &\leq n_1 \alpha^2 \lambda^2 . && \text{(B.3c)} \end{aligned}$$

■

B.3 Proof of Proposition 4.2.2

Proof. Let

$$R \triangleq \operatorname{var}\big(J_k z_2 - \widehat{\mathrm{E}}[J_k z_2 \mid V z_1]\big) - \operatorname{var}\big(J_k z_2 - \widehat{\mathrm{E}}[J_k z_2 \mid z_1]\big) \tag{B.4a}$$

$$= J_k P_{21} P_1^{-1} P_{21}^T J_k^T - J_k P_{21} V^T (V P_1 V^T)^{-1} V P_{21}^T J_k^T \,. \tag{B.4b}$$

Substitute $J_k P_{21} = \Delta + J_k P_2 H_k^T (H_k P_2 H_k^T)^{-1} H_k P_{21}$ to get

$$\begin{aligned} R = {} & \underbrace{\Delta\big(P_1^{-1} - V^T (V P_1 V^T)^{-1} V\big)\Delta^T}_{(a)} \\ & + \underbrace{\Delta\big(P_1^{-1} - V^T (V P_1 V^T)^{-1} V\big) P_{21}^T H_k^T (H_k P_2 H_k^T)^{-1} H_k P_2 J_k^T}_{(b)} \\ & + \underbrace{J_k P_2 H_k^T (H_k P_2 H_k^T)^{-1} H_k P_{21} \big(P_1^{-1} - V^T (V P_1 V^T)^{-1} V\big)\Delta^T}_{(b)^T} \\ & + \underbrace{J_k P_2 H_k^T (H_k P_2 H_k^T)^{-1} H_k P_{21} \big(P_1^{-1} - V^T (V P_1 V^T)^{-1} V\big) P_{21}^T H_k^T (H_k P_2 H_k^T)^{-1} H_k P_2 J_k^T}_{(c)} \,. \end{aligned} \tag{B.5}$$

Let

$$\Sigma \triangleq \operatorname{var}\big(H_k z_2 - \widehat{\mathrm{E}}[H_k z_2 \mid V z_1]\big) - \operatorname{var}\big(H_k z_2 - \widehat{\mathrm{E}}[H_k z_2 \mid z_1]\big) \tag{B.6a}$$

$$= H_k P_{21} P_1^{-1} P_{21}^T H_k^T - H_k P_{21} V^T (V P_1 V^T)^{-1} V P_{21}^T H_k^T \,. \tag{B.6b}$$

Note that $\bar{\varepsilon}(H_k z_2 \mid V z_1) = \operatorname{trace}(\Sigma) < \sigma^2$. Hence, the maximum eigenvalue of of Σ is bounded above by σ^2.

First consider the term marked (c) in (B.5) and make the substitution

$$H_k P_{21} V^T (V P_1 V^T)^{-1} V P_{21}^T H_k^T = H_k P_{21} P_1^{-1} P_{21}^T H_k^T - \Sigma \,. \tag{B.7}$$

Cancelling terms, we have $(c) = J_k P_2 H_k^T (H_k P_2 H_k^T)^{-1} \Sigma (H_k P_2 H_k^T)^{-1} H_k P_2 J_k^T$. Note that

$$\bar{\varepsilon}(J_k z_2 \mid V z_1) = |\operatorname{trace}(R)| \leq |\operatorname{trace}(a)| + 2|\operatorname{trace}(b)| + |\operatorname{trace}(c)| \,. \tag{B.8}$$

We will provide bounds for each of these terms. First,

$$|\operatorname{trace}(a)| = \Big|\operatorname{trace}\big([P_1^{-1} - V^T (V P_1 V^T)^{-1} V]^{T/2} \Delta^T \Delta [P_1^{-1} - V^T (V P_1 V^T)^{-1} V]^{1/2}\big)\Big| \tag{B.9a}$$

$$\leq n_1 \alpha^2 \delta^2 \,. \tag{B.9b}$$

Next,

$$|\operatorname{trace}(b)| = \big|\operatorname{trace}(\Delta\Lambda)\big| \leq \operatorname{trace}(\Delta\Delta^T)^{1/2} \operatorname{trace}(\Lambda\Lambda^T)^{1/2} \leq n_1 |\delta\lambda| \tag{B.10}$$

where we have used the Cauchy-Schwarz inequality, interpreting the trace as an inner product [93]. Finally, it is clear that $\operatorname{trace}(c) \leq n_3 \sigma^2 \beta^2$. This completes the proof. ■

Appendix C

Proofs for Chapter 5

In this appendix a discussion of (5.13) and proofs of (5.14a) (Section C.1), Proposition 5.2.1 (Section C.2), and Proposition 5.2.2 (Section C.3) are provided.

C.1 Discussion of (5.13) and Proof of (5.14a)

To see that (5.13) implies that each state depends only on its parent, consider two states $x_j(n)$ and $x_j(n+1)$ at scale j, for some even integer $n \in \{0, ..., 2^j - 2\}$. The parent of these two states is

$$x_{j-1}(n/2) = \begin{bmatrix} a_j(n/2 - \widetilde{R} + 1) \\ a_j(n/2 + \widetilde{R} - 1) \\ d_j\big(n/2 - \frac{\widetilde{R}+R}{2} + 1\big) \\ d_j\big(n/2 + \frac{\widetilde{R}+R}{2} - 1\big) \end{bmatrix} . \tag{C.1}$$

Then, for every integer $i \in \{-\widetilde{R}+1, ..., \widetilde{R}-1\}$, we have

$$a_j(n+i) = \sum_{\frac{n}{2}+\lceil\frac{i-\widetilde{R}}{2}\rceil}^{\frac{n}{2}+\lfloor\frac{i+\widetilde{R}-1}{2}\rfloor} \widetilde{h}(n+i-2p)a_{j-1}(p) + \sum_{\frac{n}{2}+\lceil\frac{i-R}{2}\rceil}^{\frac{n}{2}+\lfloor\frac{i+R-1}{2}\rfloor} \widetilde{g}(n+i-2p)d_{j-1}(p) \tag{C.2}$$

and

$$a_j(n+1+i) = \sum_{\frac{n}{2}+\lceil\frac{i+1-\widetilde{R}}{2}\rceil}^{\frac{n}{2}+\lfloor\frac{i+\widetilde{R}}{2}\rfloor} \widetilde{h}(n+i+1-2p)a_{j-1}(p)$$

$$+ \sum_{\frac{n}{2}+\lceil\frac{i+1-R}{2}\rceil}^{\frac{n}{2}+\lfloor\frac{i+R}{2}\rfloor} \widetilde{g}(n+i+1-2p)d_{j-1}(p) \, . \tag{C.3}$$

In order to check that every $a_{j-1}(p)$ and $d_{j-1}(p)$ in (C.2) and (C.3) is carried by $x_{j-1}(n/2)$, one can easily check that

$$\left[\left\lceil\frac{i-\widetilde{R}}{2}\right\rceil : \left\lfloor\frac{i+\widetilde{R}}{2}\right\rfloor\right] \subseteq [-\widetilde{R}+1 : \widetilde{R}-1] \quad \forall i \in \{-\widetilde{R}+1, ..., \widetilde{R}-1\} \tag{C.4}$$

and that

$$\left[\left\lceil\tfrac{i-R}{2}\right\rceil : \left\lfloor\tfrac{i+R}{2}\right\rfloor\right] \subset \left[-\tfrac{\widetilde{R}+R}{2}+1 : \tfrac{\widetilde{R}+R}{2}-1\right] \quad \forall i \in \{-\widetilde{R}+1, ..., \widetilde{R}-1\}. \tag{C.5}$$

Then, using (C.2) and (C.3), we get (5.14a) where, for every $j \in \{1, \dots, M-1\}$, the matrices $A_j(n)$ are $(3\widetilde{R}+R-2) \times (3\widetilde{R}+R-2)$ are defined as follows. Let

$$\ell \in [1 : 2\widetilde{R}-1], \tag{C.6a}$$

$$p_a \in \left[\left\lceil\tfrac{n+\ell-2\widetilde{R}}{2}\right\rceil : \left\lfloor\tfrac{n+\ell-1}{2}\right\rfloor\right], \tag{C.6b}$$

$$p_d \in \left[\left\lceil\tfrac{n+\ell-\widetilde{R}-R}{2}\right\rceil : \left\lfloor\tfrac{n+\ell+R-\widetilde{R}-1}{2}\right\rfloor\right]. \tag{C.6c}$$

Then,

$$A_j(n)\big(\ell, p_a - \lfloor n/2 \rfloor + \widetilde{R}\big) = \widetilde{h}(n+\ell-\widetilde{R}-2p_a), \tag{C.7a}$$

$$A_j(n)\Big(\ell, p_d - \lfloor n/2 \rfloor + \tfrac{5\widetilde{R}+1}{2} - 1\Big) = \widetilde{g}(n+\ell-\widetilde{R}-2p_d). \tag{C.7b}$$

When $j = M$, $A_M(n)$ are vectors of length $3\widetilde{R}+R-2$ and are defined as follows. Let

$$p_a \in \left[\left\lceil\tfrac{n+1-2\widetilde{R}}{2}\right\rceil : \left\lfloor\tfrac{n}{2}\right\rfloor\right], \tag{C.8a}$$

$$p_d \in \left[\left\lceil\tfrac{n+1-\widetilde{R}-R}{2}\right\rceil : \left\lfloor\tfrac{n+R-\widetilde{R}}{2}\right\rfloor\right]. \tag{C.8b}$$

Then,

$$A_M(n)\big(1, p_a - \lfloor n/2 \rfloor + \widetilde{R}\big) = \widetilde{h}(n+1-\widetilde{R}-2p_a), \tag{C.9a}$$

$$A_M(n)\Big(1, p_d - \lfloor n/2 \rfloor + \tfrac{5\widetilde{R}+1}{2} - 1\Big) = \widetilde{g}(n+1-\widetilde{R}-2p_d). \tag{C.9b}$$

■ C.2 Proof of Proposition 5.2.1

The proof of Proposition 5.2.1 relies on the following lemma.

Lemma C.2.1. *Let i be an integer in $\{1, \dots, \frac{\widetilde{R}+R}{2}-1\}$, then,*

$$\sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R}-2p)a_j(n - \tfrac{\widetilde{R}-R}{2} - i + p) + \widetilde{g}(R-2p)d_j(n-i+p)$$

$$= \sum_{k=-\widetilde{R}+1}^{\widetilde{R}-2} \alpha_i(k)a_{j+1}(2n+k) \tag{C.10}$$

and

$$\sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R}+1+2p)a_j(n+\tfrac{\widetilde{R}-R}{2}+i-p)+\widetilde{g}(-R+1+2p)d_j(n+i-p)$$
$$= \sum_{k=-\widetilde{R}+3}^{\widetilde{R}} \beta_i(k)a_{j+1}(2n+k) \quad \text{(C.11)}$$

where

$$\alpha_i(k) = \begin{cases} \sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R}-2p)h(k+2i-2p+\widetilde{R}-R)+\widetilde{g}(R-2p)g(k+2i-2p) & \textit{if } k \leq 2R-\widetilde{R}-2\,, \\ \sum_{p=0}^{i-1} \widetilde{g}(R-2p)g(k+2i-2p) & \textit{if } k > 2R-\widetilde{R}-2 \end{cases} \quad \text{(C.12)}$$

and

$$\beta_i(k) =$$
$$\begin{cases} \sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R}+1+2p)h(k+2p-2i+R-\widetilde{R})+\widetilde{g}(-R+1+2p)g(k+2p-2i) & \textit{if } k \geq \widetilde{R}-2R+3\,, \\ \sum_{p=0}^{i-1} \widetilde{g}(-R+1+2p)g(k+2p-2i) & \textit{if } k < \widetilde{R}-2R+3\,. \end{cases} \quad \text{(C.13)}$$

Proof. We will first show (C.10) holds. Using the wavelet decomposition algorithm (5.7) we have

$$a_j\big(n-\tfrac{\widetilde{R}-R}{2}-i+p\big) = \sum_{k=2n-2i+2p-\widetilde{R}+1}^{2n+2R-\widetilde{R}-2i+2p} h(k-2n+\widetilde{R}-R+2i-2p)a_{j+1}(k) \quad \text{(C.14a)}$$
$$= \sum_{u=-2i+2p-\widetilde{R}+1}^{-2i+2p+2R-\widetilde{R}} h(u+2i-2p+\widetilde{R}-R)a_{j+1}(u+2n) \quad \text{(C.14b)}$$
$$= \sum_{u=-2i-\widetilde{R}+1}^{2R-\widetilde{R}-2} h(u+2i-2p+\widetilde{R}-R)a_{j+1}(2n+u) \quad \text{(C.14c)}$$

where (C.14b) is obtained by making the change of variables $k = u+2n$ and where (C.14c) follows from the fact that

$$u < -2i+2p-\widetilde{R}+1 \Rightarrow u+2i-2p+\widetilde{R}-R < -R+1 \quad \text{(C.15a)}$$
$$\Rightarrow h(u+2i-2p+\widetilde{R}-R) = 0 \quad \text{(C.15b)}$$

and

$$u > -2i+2p+2R-\widetilde{R} \Rightarrow u+2i-2p+\widetilde{R}-R > R \quad \text{(C.16a)}$$
$$\Rightarrow h(u+2i-2p+\widetilde{R}-R) = 0\,. \quad \text{(C.16b)}$$

Therefore,

$$\sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R}-2p)a_j\big(n-\tfrac{\widetilde{R}-R}{2}-i+p\big) = \sum_{k=-2i-\widetilde{R}+1}^{2R-\widetilde{R}-2} \Big(\sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R}-2p)h(k+2i-2p+\widetilde{R}-R)\Big)a_{j+1}(2n+k)\,. \quad \text{(C.17)}$$

Using again the wavelet decomposition algorithm we have

$$\begin{aligned} d_j(n-i+p) &= \sum_{k=2n-2i+2p-\widetilde{R}+1}^{2n-2i+2p+\widetilde{R}} g(k-2n+2i-2p)a_{j+1}(k) && \text{(C.18a)} \\ &= \sum_{u=-2i+2p-\widetilde{R}+1}^{-2i+2p+\widetilde{R}} g(u+2i-2p)a_{j+1}(u+2n) && \text{(C.18b)} \\ &= \sum_{u=-2i-\widetilde{R}+1}^{\widetilde{R}-2} g(u+2i-2p)a_{j+1}(2n+u) && \text{(C.18c)} \end{aligned}$$

where (C.18b) is obtained by making the change of variables $k = u + 2n$ and (C.18c) follows from the fact that

$$u < -2i+2p-\widetilde{R}+1 \Rightarrow u+2i-2p < -\widetilde{R}+1 \Rightarrow g(u+2i-2p) = 0 \quad \text{(C.19)}$$

and

$$u > -2i+2p+\widetilde{R} \Rightarrow u+2i-2p > \widetilde{R} \Rightarrow g(u+2i-2p) = 0\,. \quad \text{(C.20)}$$

Therefore,

$$\sum_{p=0}^{i-1} \widetilde{g}(R-2p)d_j(n-i+p) = \sum_{k=-2i-\widetilde{R}+1}^{\widetilde{R}-2} \Big(\sum_{p=0}^{i-1} \widetilde{g}(R-2p)g(k+2i-2p)\Big)a_{j+1}(2n+k)\,. \quad \text{(C.21)}$$

Using (C.17) and (C.21) and the fact that $\widetilde{R}-2 \geq 2R-\widetilde{R}-2$ we have

$$\sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R}-2p)a_j\big(n-\tfrac{\widetilde{R}-R}{2}-i+p\big) + \widetilde{g}(R-2p)d_j(n-i+p) = \sum_{k=-2i-\widetilde{R}+1}^{\widetilde{R}-2} \alpha_i(k)a_{j+1}(2n+k) \quad \text{(C.22)}$$

where $\alpha_i(k)$ is as in (C.12).

To prove (C.10), we need to show that $\alpha_i(k) = 0$ for $k \in [-2i - \widetilde{R} + 1 : -\widetilde{R}]$. Notice that when $k \leq -\widetilde{R}$ then $k \leq 2R - \widetilde{R} - 2$ so

$$\alpha_i(k) = \sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R} - 2p)h(k + 2i - 2p + \widetilde{R} - R) + \widetilde{g}(R - 2p)g(k + 2i - 2p)\,. \tag{C.23}$$

Using (5.6) we have

$$\widetilde{g}(R - 2p)g(k + 2i - 2p) = (-1)^{k+R}h(1 - R + 2p)\widetilde{h}(1 + 2p - 2i - k)\,. \tag{C.24}$$

We are going to distinguish between the case where $k + R$ is odd and $k + R$ is even. Notice that $k + R$ is even (respectively odd) if and only if $k + \widetilde{R}$ is even (respectively odd) since R and $\widetilde{R}$ have the same parity.

Case 1: $k + \widetilde{R}$ is odd. In this case we have

$$\alpha_i(k) = \sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R} - 2p)h(k + 2i - 2p + \widetilde{R} - R) - h(1 - R + 2p)\widetilde{h}(1 + 2p - 2i - k)\,. \tag{C.25}$$

Recall that $k \in [-2i - \widetilde{R} + 1 : -\widetilde{R}]$. We can therefore write $k = -\widetilde{R} - 2i + 2\ell + 1$ where $\ell = 0, 1, \ldots, i - 1$. Proving that $\alpha_i(k) = 0$ is equivalent to showing that

$$\sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R} - 2p)h(2\ell - 2p - R + 1) - h(1 - R + 2p)\widetilde{h}(2p - 2\ell + \widetilde{R}) = 0 \tag{C.26}$$

where the left-hand side follows from (C.25) with the change of variables $k = -\widetilde{R} - 2i + 2\ell + 1$. Continuing, we have

$$\begin{aligned}&\sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R} - 2p)h(2\ell - 2p - R + 1) - h(1 - R + 2p)\widetilde{h}(2p - 2\ell + \widetilde{R}) \\ &\quad = \sum_{p=0}^{\ell} \widetilde{h}(\widetilde{R} - 2p)h(2\ell - 2p - R + 1) - \sum_{p=0}^{\ell} h(1 - R + 2p)\widetilde{h}(2p - 2\ell + \widetilde{R})\,.\end{aligned} \tag{C.27}$$

The equality in (C.27) holds because

$$p > \ell \Rightarrow 2\ell - 2p - R + 1 < -R + 1 \Rightarrow h(2\ell - 2p - R + 1) = 0 \tag{C.28}$$

and

$$p > \ell \Rightarrow \widetilde{R} - 2\ell + 2p > \widetilde{R} \Rightarrow \widetilde{h}(\widetilde{R} - 2\ell + 2p) = 0\,. \tag{C.29}$$

Consider now the second term of (C.27). By making the change of variables $p = -u + \ell$ we have

$$\sum_{p=0}^{\ell} h(1 - R + 2p)\widetilde{h}(2p - 2\ell + \widetilde{R}) = \sum_{u=0}^{\ell} h(2\ell - 2u - R + 1)\widetilde{h}(\widetilde{R} - 2u)\,. \tag{C.30}$$

Therefore, from (C.27) we conclude that $\alpha_i(k) = 0$ for every $k \in [-2i - \widetilde{R} + 1 : -\widetilde{R}]$ such that $k + \widetilde{R}$ is odd.

Case 2: $k + \widetilde{R}$ is even. In this case we have

$$\alpha_i(k) = \sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R} - 2p)h(k + 2i - 2p + \widetilde{R} - R) + h(1 - R + 2p)\widetilde{h}(1 + 2p - 2i - k)\,. \tag{C.31}$$

Let us write $k = -\widetilde{R} - 2i + 2\ell$ where $\ell = 1, \dots, i$. Then, showing that $\alpha_i(k) = 0$ is equivalent to showing that

$$\sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R} - 2p)h(2\ell - 2p - R) + h(1 - R + 2p)\widetilde{h}(2p - 2\ell + \widetilde{R} + 1) = 0\,. \tag{C.32}$$

Continuing, we have

$$\begin{aligned}
&\sum_{p=0}^{i-1} \widetilde{h}(\widetilde{R} - 2p)h(2\ell - 2p - R) + h(1 - R + 2p)\widetilde{h}(2p - 2\ell + \widetilde{R} + 1) \\
&\quad = \underbrace{\sum_{p=0}^{\ell-1} \widetilde{h}(\widetilde{R} - 2p)h(2\ell - 2p - R)}_{y} + \underbrace{\sum_{p=0}^{\ell-1} h(1 - R + 2p)\widetilde{h}(2p - 2\ell + \widetilde{R} + 1)}_{z}\,.
\end{aligned} \tag{C.33}$$

The equality in (C.33) holds because

$$p \geq \ell \Rightarrow 2\ell - 2p - R \leq -R \Rightarrow h(2\ell - 2p - R) = 0 \tag{C.34}$$

and

$$p \geq \ell \Rightarrow 1 + \widetilde{R} - 2\ell + 2p \geq \widetilde{R} + 1 \Rightarrow \widetilde{h}(1 + \widetilde{R} - 2\ell + 2p) = 0\,. \tag{C.35}$$

We have

$$y = \sum_{\substack{v=0 \\ v \text{ even}}}^{2\ell-2} \widetilde{h}(\widetilde{R} - v)h(2\ell - v - R) = \sum_{\substack{u=1 \\ u \text{ even}}}^{2\ell} \widetilde{h}(\widetilde{R} + u - 2\ell)h(-R + u) \tag{C.36}$$

where the last equality is obtained by the change of variables $2\ell - v = u$. We also have

$$z = \sum_{\substack{v=1 \\ v \text{ odd}}}^{2\ell} \widetilde{h}(\widetilde{R} + v - 2\ell)h(v - R)\,. \tag{C.37}$$

Therefore,

$$y + z = \sum_{v=1}^{2\ell} h(-R + v)\widetilde{h}(\widetilde{R} + v - 2\ell)\,. \tag{C.38}$$

Applying (5.4) with $n = \ell - \frac{\widetilde{R}+R}{2}$ we get

$$\sum_{k=-R+1}^{-R+2\ell} h(k)\widetilde{h}(k + R + \widetilde{R} - 2\ell) = 0 \tag{C.39}$$

because $n \neq 0$. Therefore, by making the change of variables $k = -R + v$,

$$\sum_{v=1}^{2\ell} h(-R + v)\widetilde{h}(\widetilde{R} + v - 2\ell) = 0\,. \tag{C.40}$$

Hence, $\alpha_i(k) = 0$ is zero for every $k \in [-2i - \widetilde{R} + 1 : -\widetilde{R}]$ such that $k + \widetilde{R}$ is even. This concludes the proof of (C.10).

We now show that (C.11) holds. Using the wavelet decomposition algorithm (5.7) we have

$$a_j\big(n + \tfrac{\widetilde{R}-R}{2} + i - p\big) = \sum_{k=2n+2i-2p+\widetilde{R}-2R+1}^{2n+\widetilde{R}+2i-2p} h(k - 2n - \widetilde{R} + R - 2i + 2p)a_{j+1}(k) \tag{C.41a}$$

$$= \sum_{u=2i-2p+\widetilde{R}-2R+1}^{2i-2p+\widetilde{R}} h(u - 2i + 2p - \widetilde{R} + R)a_{j+1}(u + 2n) \tag{C.41b}$$

$$= \sum_{u=\widetilde{R}-2R+3}^{2i+\widetilde{R}} h(u - 2i + 2p - \widetilde{R} + R)a_{j+1}(2n + u) \tag{C.41c}$$

where (C.41b) is obtained by making the change of variables $k = u + 2n$ and where (C.41c) follows from the fact that

$$u < 2i - 2p + \widetilde{R} - 2R + 1 \Rightarrow u - 2i + 2p - \widetilde{R} + R \leq -R \tag{C.42a}$$

$$\Rightarrow h(u - 2i + 2p - \widetilde{R} + R) = 0 \tag{C.42b}$$

and

$$u > 2i - 2p + \widetilde{R} \Rightarrow u - 2i + 2p - \widetilde{R} + R > R \Rightarrow h(u - 2i + 2p - \widetilde{R} + R) = 0\,. \quad \text{(C.43)}$$

Therefore,

$$\sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R} + 1 - 2p) a_j\big(n + \tfrac{\widetilde{R}-R}{2} + i - p\big)$$
$$= \sum_{k=\widetilde{R}-2R+3}^{2i+\widetilde{R}} \Big(\sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R} + 1 + 2p) h(k - 2i + 2p - \widetilde{R} + R)\Big) a_{j+1}(2n + k)\,. \quad \text{(C.44)}$$

Using again the wavelet decomposition algorithm we have

$$d_j(n + i - p) = \sum_{k=2n+2i-2p-\widetilde{R}+1}^{2n+2i-2p+\widetilde{R}} g(k - 2n - 2i + 2p) a_{j+1}(k) \quad \text{(C.45a)}$$
$$= \sum_{u=2i-2p-\widetilde{R}+1}^{2i-2p+\widetilde{R}} g(u - 2i + 2p) a_{j+1}(u + 2n) \quad \text{(C.45b)}$$
$$= \sum_{u=-\widetilde{R}+3}^{2i+\widetilde{R}} g(u - 2i + 2p) a_{j+1}(2n + u) \quad \text{(C.45c)}$$

where (C.45b) is obtained by making the change of variables $k = u + 2n$ and (C.45c) follows from the fact that

$$u < 2i - 2p - \widetilde{R} + 1 \Rightarrow u - 2i + 2p < -\widetilde{R} + 1 \Rightarrow g(u - 2i + 2p) = 0 \quad \text{(C.46)}$$

and

$$u > 2i - 2p + \widetilde{R} \Rightarrow u - 2i + 2p > \widetilde{R} \Rightarrow g(u - 2i + 2p) = 0\,. \quad \text{(C.47)}$$

Therefore,

$$\sum_{p=0}^{i-1} \widetilde{g}(-R + 2p + 1) d_j(n + i - p)$$
$$= \sum_{k=-\widetilde{R}+3}^{2i+\widetilde{R}} \Big(\sum_{p=0}^{i-1} \widetilde{g}(-R + 1 + 2p) g(k - 2i + 2p)\Big) a_{j+1}(2n + k)\,. \quad \text{(C.48)}$$

Using (C.44) and (C.48) and the fact that $\widetilde{R}+3 \leq \widetilde{R}-2R+3$ we have

$$\sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R}+1+2p)a_j\big(n+\tfrac{\widetilde{R}-R}{2}+i-p\big) \\ +\widetilde{g}(-R+1+2p)d_j(n+i-p) = \sum_{k=-\widetilde{R}+3}^{2i+\widetilde{R}} \beta_i(k)a_{j+1}(2n+k) \quad \text{(C.49)}$$

where $\beta_i(k)$ is as in (C.13).

To prove (C.11), we need to show that $\beta_i(k)=0$ for $k \in [\widetilde{R}+1 : 2i+\widetilde{R}]$. Notice that when $k \geq \widetilde{R}+1$ then $k \geq \widetilde{R}-2R+3$ so

$$\beta_i(k) = \sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R}+1+2p)h(k+2p-2i-\widetilde{R}+R) + \widetilde{g}(-R+1+2p)g(k+2p-2i)\,. \quad \text{(C.50)}$$

Using (5.6) we have

$$\widetilde{g}(-R+1+2p)g(k-2i+2p) = (-1)^{-k+R+1}h(R-2p)\widetilde{h}(1-k-2p+2i)\,. \quad \text{(C.51)}$$

We are going to distinguish between the case where $-k+R$ is even and $-k+R$ is odd. Notice that $-k+R$ is even (respectively odd) if and only if $-k+\widetilde{R}$ is even (respectively odd) since R and $\widetilde{R}$ have the same parity.

Case 1: $-k+\widetilde{R}$ is even. In this case we have

$$\beta_i(k) = \sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R}+1+2p)h(k-2i+2p-\widetilde{R}+R) - h(R-2p)\widetilde{h}(1+2i-2p-k)\,. \quad \text{(C.52)}$$

Recall that $k \in [\widetilde{R}+1 : 2i+\widetilde{R}]$. We can therefore write $-k = -\widetilde{R}-2i+2\ell$ where $\ell = 0,1,\dots,i-1$. Therefore, showing that $\beta_i(k)=0$ is equivalent to showing that

$$\sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R}+1+2p)h(2p-2\ell+R) - h(R-2p)\widetilde{h}(2\ell-2p-\widetilde{R}+1) = 0\,. \quad \text{(C.53)}$$

Continuing, we have

$$\sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R}+1+2p)h(2p-2\ell+R) - h(R-2p)\widetilde{h}(2\ell-2p-\widetilde{R}+1) \\ = \sum_{p=0}^{\ell} \widetilde{h}(-\widetilde{R}+2p+1)h(2p-2\ell+R) - \sum_{p=0}^{\ell} h(R-2p)\widetilde{h}(2\ell-2p-\widetilde{R}+1)\,. \quad \text{(C.54)}$$

The equality in (C.54) holds because

$$p > \ell \Rightarrow 2p - 2\ell + R > R \Rightarrow h(2p - 2\ell + R) = 0 \tag{C.55}$$

and

$$p > \ell \Rightarrow 2\ell - 2p - \widetilde{R} + 1 < -\widetilde{R} + 1 \Rightarrow \widetilde{h}(2\ell - 2p - \widetilde{R} + 1) = 0\,. \tag{C.56}$$

Consider now the second term of (C.33). By making the change of variables $p = -u + \ell$ we have

$$\sum_{p=0}^{\ell} h(R - 2p)\widetilde{h}(2\ell - 2p - \widetilde{R} + 1) = \sum_{u=0}^{\ell} \widetilde{h}(2u - \widetilde{R} + 1)h(R + 2u - 2\ell)\,. \tag{C.57}$$

Therefore, from (C.33) we conclude that $\beta_i(k) = 0$ for every $k \in [\widetilde{R} + 1 : 2i + \widetilde{R}]$ such that $-k + \widetilde{R}$ is even.

Case 2: $-k + \widetilde{R}$ is odd. In this case we have

$$\beta_i(k) = \sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R} + 1 + 2p)h(k - 2i + 2p - \widetilde{R} + R) + h(R - 2p)\widetilde{h}(1 - 2p + 2i - k)\,. \tag{C.58}$$

Let us write $-k = -\widetilde{R} - 2i + 2\ell - 1$ where $\ell = 1, \dots, i$. Showing that $\beta_i(k) = 0$ is, therefore, equivalent to showing

$$\sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R} + 2p + 1)h(2p - 2\ell + R + 1) + h(R - 2p)\widetilde{h}(2\ell - 2p - \widetilde{R})\,. \tag{C.59}$$

Continuing, we have

$$\begin{aligned} &\sum_{p=0}^{i-1} \widetilde{h}(-\widetilde{R} + 2p + 1)h(2p - 2\ell + R + 1) + h(R - 2p)\widetilde{h}(2\ell - 2p - \widetilde{R}) \\ &= \underbrace{\sum_{p=0}^{\ell-1} \widetilde{h}(-\widetilde{R} + 2p + 1)h(2p - 2\ell + R + 1)}_{y} + \underbrace{\sum_{p=0}^{\ell-1} h(R - 2p)\widetilde{h}(2\ell - 2p - \widetilde{R})}_{z}\,. \end{aligned} \tag{C.60}$$

The equality in (C.60) holds because

$$p \geq \ell \Rightarrow 2p - 2\ell + R + 1 \geq R + 1 \Rightarrow h(2p - 2\ell + R + 1) = 0 \tag{C.61}$$

and

$$p \geq \ell \Rightarrow 2\ell - 2p - \widetilde{R} \leq -\widetilde{R} \Rightarrow \widetilde{h}(2\ell - 2p - \widetilde{R}) = 0\,. \tag{C.62}$$

We have

$$y = \sum_{\substack{v=1 \\ v \text{ odd}}}^{2\ell} \widetilde{h}(-\widetilde{R}+v)h(-2\ell+v+R)\,. \tag{C.63}$$

We also have

$$z = \sum_{\substack{v=0 \\ v \text{ even}}}^{2\ell-2} \widetilde{h}(-\widetilde{R}-v+2\ell)h(R-v) = \sum_{\substack{u=1 \\ u \text{ even}}}^{2\ell} \widetilde{h}(-\widetilde{R}+u)h(R+u-2\ell) \tag{C.64}$$

where in the last equality we have made the change of variables $u = 2\ell - v$.

Therefore,

$$y + z = \sum_{v=1}^{2\ell} \widetilde{h}(-\widetilde{R}+v)h(R+v-2\ell)\,. \tag{C.65}$$

Applying (5.4) with $n = \ell - \frac{\widetilde{R}+R}{2}$ we get

$$\sum_{k=-\widetilde{R}+1}^{-\widetilde{R}+2\ell} \widetilde{h}(k)h(k+R+\widetilde{R}-2\ell) = 0 \tag{C.66}$$

because $n \neq 0$. Therefore, by making the change of variables $k = -\widetilde{R}+v$, we have

$$\sum_{v=1}^{2\ell} \widetilde{h}(-\widetilde{R}+v)h(R+v-2\ell) = 0\,. \tag{C.67}$$

Hence, $\beta_i(k) = 0$ for every $k \in [\widetilde{R}+1, 2i+\widetilde{R}]$ such that $-k+\widetilde{R}$ is odd. This concludes the proof of Lemma C.2.1. ■

We now prove Proposition 5.2.1.

Proof of Proposition 5.2.1. Define

$$K_1 \triangleq \begin{pmatrix} \alpha_{\frac{\widetilde{R}+R}{2}-1}(-\widetilde{R}+1) & \alpha_{\frac{\widetilde{R}+R}{2}-1}(-\widetilde{R}+2) & \ldots & \alpha_{\frac{\widetilde{R}+R}{2}-1}(\widetilde{R}-2) \\ \alpha_{\frac{\widetilde{R}+R}{2}-2}(-\widetilde{R}+1) & \alpha_{\frac{\widetilde{R}+R}{2}-2}(-\widetilde{R}+2) & \ldots & \alpha_{\frac{\widetilde{R}+R}{2}-2}(\widetilde{R}-2) \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1(-\widetilde{R}+1) & \alpha_1(-\widetilde{R}+2) & \ldots & \alpha_1(\widetilde{R}-2) \end{pmatrix}, \tag{C.68a}$$

$$K_2 \triangleq \begin{pmatrix} \beta_1(-\widetilde{R}+3) & \beta_1(-\widetilde{R}+4) & \ldots & \beta_1(\widetilde{R}) \\ \beta_2(-\widetilde{R}+3) & \beta_2(-\widetilde{R}+4) & \ldots & \beta_2(\widetilde{R}) \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{\frac{\widetilde{R}+R}{2}-1}(-\widetilde{R}+3) & \beta_{\frac{\widetilde{R}+R}{2}-1}(-\widetilde{R}+4) & \ldots & \beta_{\frac{\widetilde{R}+R}{2}-1}(\widetilde{R}) \end{pmatrix} \tag{C.68b}$$

and the $\big(\frac{\widetilde{R}+R}{2}-1\big) \times \big(\frac{\widetilde{R}+R}{2}-1\big)$ triangular matrices H_1, H_2, G_1, G_2 as

$$H_1(\ell, c) \triangleq \begin{cases} \widetilde{h}(\widetilde{R}-2(c-\ell)) & \text{for } \ell \in \big[1 : \frac{\widetilde{R}+R}{2}-1\big],\ c \in \big[\ell : \frac{\widetilde{R}+R}{2}-1\big], \\ 0 & \text{otherwise}, \end{cases} \tag{C.69a}$$

$$G_1(\ell, c) \triangleq \begin{cases} \widetilde{g}(R-2(c-\ell)) & \text{for } \ell \in \big[1 : \frac{\widetilde{R}+R}{2}-1\big],\ c \in \big[\ell : \frac{\widetilde{R}+R}{2}-1\big], \\ 0 & \text{otherwise}, \end{cases} \tag{C.69b}$$

$$H_2(\ell, c) \triangleq \begin{cases} \widetilde{h}(-\widetilde{R}+1+2(\ell-c)) & \text{for } \ell \in \big[1 : \frac{\widetilde{R}+R}{2}-1\big],\ c \in \big[1 : \ell\big], \\ 0 & \text{otherwise}, \end{cases} \tag{C.69c}$$

$$G_2(\ell, c) \triangleq \begin{cases} \widetilde{g}(-R+1+2(\ell-c)) & \text{for } \ell \in \big[1 : \frac{\widetilde{R}+R}{2}-1\big],\ c \in \big[1 : \ell\big], \\ 0 & \text{otherwise}. \end{cases} \tag{C.69d}$$

Then (C.10) and (C.11) imply that

$$G_1 \begin{bmatrix} d_j\big(n-\frac{\widetilde{R}+R}{2}+1\big) \\ \vdots \\ d_j(n-1) \end{bmatrix} + H_1 \begin{bmatrix} a_j(n-\widetilde{R}+1) \\ \vdots \\ a_j\big(n-\frac{\widetilde{R}-R}{2}-1\big) \end{bmatrix} = K_1 \begin{bmatrix} a_{j+1}(2n-\widetilde{R}+1) \\ \vdots \\ a_{j+1}(2n+\widetilde{R}-2) \end{bmatrix}, \tag{C.70a}$$

$$G_2 \begin{bmatrix} d_j(n+1) \\ \vdots \\ d_j\big(n+\frac{\widetilde{R}+R}{2}-1\big) \end{bmatrix} + H_2 \begin{bmatrix} a_j\big(n+\frac{\widetilde{R}-R}{2}+1\big) \\ \vdots \\ a_j(n+\widetilde{R}-1) \end{bmatrix} = K_2 \begin{bmatrix} a_{j+1}(2n-\widetilde{R}+3) \\ \vdots \\ a_{j+1}(2n+\widetilde{R}) \end{bmatrix}. \tag{C.70b}$$

Since $\widetilde{g}(R) \neq 0$ and $\widetilde{g}(-R+1) \neq 0$, then G_1 and G_2 are invertible and (5.15a) and (5.15b) follow with $L_1 = -G_1^{-1}H_1$, $J_1 = G_1^{-1}K_1$, $L_2 = -G_2^{-1}H_2$, $J_2 = G_2^{-1}K_2$. ■

■ C.3 Proof of Proposition 5.2.2

Proof. First, let us define for notational simplicity $\widehat{R} \triangleq \frac{\widetilde{R}+R}{2}$. Now consider the children $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$ of $x_j(n)$ which are defined as

$$x_{j+1}(2n) \triangleq \begin{bmatrix} a_{j+1}(2n-\widetilde{R}+1) \\ \vdots \\ a_{j+1}(2n-1) \\ a_{j+1}(2n) \\ a_{j+1}(2n+1) \\ \vdots \\ a_{j+1}(2n+\widetilde{R}-1) \\ d_{j+1}(2n-\widehat{R}+1) \\ \vdots \\ d_{j+1}(2n-1) \\ d_{j+1}(2n) \\ d_{j+1}(2n+1) \\ \vdots \\ d_{j+1}(2n+\widehat{R}-1) \\ \xi_{j+1}(2n) \end{bmatrix} \tag{C.71a}$$

and

$$x_{j+1}(2n+1) \triangleq \begin{bmatrix} a_{j+1}(2n-\widetilde{R}+2) \\ \vdots \\ a_{j+1}(2n) \\ a_{j+1}(2n+1) \\ a_{j+1}(2n+2) \\ \vdots \\ a_{j+1}(2n+\widetilde{R}) \\ d_{j+1}(2n-\widehat{R}+2) \\ \vdots \\ d_{j+1}(2n) \\ d_{j+1}(2n+1) \\ d_{j+1}(2n+2) \\ \vdots \\ d_{j+1}(2n+\widehat{R}) \\ \xi_{j+1}(2n+1) \end{bmatrix} \tag{C.71b}$$

where

$$\xi_{j+1}(2n) = \left[\xi_j^1(n) \quad a_j(n-\widetilde{R}+1) \quad \cdots \quad a_j\left(n - \tfrac{\widetilde{R}-R}{2} - 1\right)\right]^T \tag{C.72a}$$

and

$$\xi_{j+1}(2n+1) = \left[\xi_j^2(n) \quad a_j\left(n + \tfrac{\widetilde{R}-R}{2} + 1\right) \quad \cdots \quad a_j(n+\widetilde{R}-1)\right]^T . \tag{C.72b}$$

Then we have the following:

- $\left[a_j(n-\widetilde{R}+1) \quad \cdots \quad a_j\left(n - \tfrac{\widetilde{R}-R}{2} - 1\right)\right]^T$ is clearly a linear function of $x_{j+1}(2n)$ since it is simply copied in $\xi_{j+1}(2n)$.

- $\left[a_j\left(n + \tfrac{\widetilde{R}-R}{2} + 1\right) \quad \cdots \quad a_j(n+\widetilde{R}-1)\right]^T$ is clearly a linear function of $x_{j+1}(2n+1)$ since it is simply copied in $\xi_{j+1}(2n+1)$.

- Using Proposition 5.2.1, $\left[d_j(n-\widehat{R}+1) \quad \cdots \quad d_j(n-1)\right]^T$ is a linear function of $x_{j+1}(2n)$.

- Using Proposition 5.2.1, $\left[d_j(n+1) \quad \cdots \quad d_j(n+\widehat{R}-1)\right]^T$ is a linear function of $x_{j+1}(2n+1)$.

- The wavelet decomposition formulas (5.7) imply that $d_j(n)$ is a linear function of $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$ since they contain $a_{j+1}(m)$ for $m \in \{2n-\widetilde{R}+1, \ldots, 2n+\widetilde{R}\}$.

- The wavelet decomposition formulas (5.7) also imply that

$$\left[a_j\left(n - \tfrac{\widetilde{R}-R}{2}\right) \quad \cdots \quad a_j\left(n + \tfrac{\widetilde{R}-R}{2}\right)\right]^T \tag{C.73}$$

is a linear function of $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$. Indeed, for $i \in \mathcal{I} \triangleq \left[-\frac{\widetilde{R}-R}{2} : \frac{\widetilde{R}-R}{2}\right]$ (5.7) implies that

$$a_j(n+i) = \sum_{p=2n+2i-R+1}^{2n+2i+R} h(p-2n-2i)a_{j+1}(p) . \tag{C.74}$$

Since $\{2n+2i-R+1, \ldots, 2n+2i+R\}_{i\in\mathcal{I}} = \{2n-\widetilde{R}+1, \ldots, 2n+\widetilde{R}\}$, it follows that the vector in (C.73) is a linear function of $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$.

- Finally, $\xi_j(n)$ is a linear function of $x_{j+1}(2n)$ and $x_{j+1}(2n+1)$ since the two parts $\xi_j^1(n)$ and $\xi_j^2(n)$ that compose $\xi_j(n)$ are carried by $\xi_{j+1}(2n)$ and $\xi_{j+1}(2n+1)$, respectively.

■

Appendix D

Proofs for Chapter 6

In this appendix we provide proofs for Proposition 6.4.1 (Section D.1), Proposition 6.5.1 (Section D.2), Proposition 6.5.2 (Section D.3), Proposition 6.5.3 (Section D.4), and a sketch of a proof of Proposition 6.5.4 (Section D.5).

■ D.1 Proof of Proposition 6.4.1

Proof. Using (6.24a) which states that

$$\begin{bmatrix} 0 & B_U^b \end{bmatrix} P_{Q^2} = \begin{bmatrix} \delta_U^{a,b} & 0 & \cdots & 0 & \varepsilon_U^b \end{bmatrix} , \tag{D.1}$$

we have

$$\left(\begin{array}{c|c} \begin{matrix} & & \\ & I & \\ & & \end{matrix} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline 0 & B_U^b \end{array}\right) P_{Q^2} \left(\begin{array}{c|c} \begin{matrix} & & \\ & I & \\ & & \end{matrix} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline 0 & B_U^b \end{array}\right)^T = \underbrace{\left(\begin{array}{ccc|c} & & & \delta_U^{a,b} \\ & P_{U_a^2} & & \vdots \\ & & & 0 \\ \hline \delta_U^{a,b} & 0 & \cdots \; 0 & \varepsilon_U^b \end{array}\right)}_{K} . \tag{D.2}$$

By Sylvester's law of inertia [84], we conclude that $P_{Q^2} > 0$ if and only if the matrix on the right-hand side of (D.2), which we have denoted by K, is positive-definite. The identity

$$\begin{pmatrix} F_1 & F_2 \\ F_2^T & F_3 \end{pmatrix} = \begin{pmatrix} I & 0 \\ F_2^T F_1^{-1} & I \end{pmatrix} \begin{pmatrix} F_1 & 0 \\ 0 & F_3 - F_2^T F_1^{-1} F_2 \end{pmatrix} \begin{pmatrix} I & 0 \\ F_2^T F_1^{-1} & I \end{pmatrix}^T \tag{D.3}$$

implies that $\left(\begin{smallmatrix} F_1 & F_2 \\ F_2^T & F_3 \end{smallmatrix}\right) > 0$ if and only if $F_1 > 0$ and $F_3 - F_2^T F_1^{-1} F_2 > 0$. Applying this fact to K and using the fact that $P_{U_a^2} > 0$, we conclude that $P_{Q^2} > 0$ if and only if

$$\varepsilon_U^b - (\delta_U^{a,b})^2 \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} P_{U_a^2}^{-1} \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T > 0 . \tag{D.4}$$

The positivity condition (D.4) can be simplified by using (6.27) which states that

$$\underbrace{\begin{bmatrix} 1 & -L_U^a \end{bmatrix}}_{A_U^a} P_{U_a^2} = \begin{bmatrix} \varepsilon_U^a & 0 & \cdots & 0 \end{bmatrix} . \tag{D.5}$$

Therefore,

$$\begin{aligned} 1 &= A_U^a \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T && \text{(D.6a)} \\ &= \varepsilon_U^a \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} P_{U_a^2}^{-1} \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T && \text{(D.6b)} \end{aligned}$$

and (D.4) simplifies to

$$\varepsilon_U^b - \left(\delta_U^{a,b}\right)^2 \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} P_{U_a^2}^{-1} \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T = \varepsilon_U^b - \frac{\left(\delta_U^{a,b}\right)^2}{\varepsilon_U^a} > 0 \tag{D.7}$$

or, using the fact that ε_U^a is positive,

$$\left(\delta_U^{a,b}\right)^2 < \varepsilon_U^a \varepsilon_U^b \,. \tag{D.8}$$

Thus, by the definition of $\rho_U^{a,b}$ (cf., (6.32)),

$$\left|\rho_U^{a,b}\right| = \left|\frac{\delta_U^{a,b}}{\sqrt{\varepsilon_U^a \varepsilon_U^b}}\right| < 1 \,. \tag{D.9}$$

That the choice of $\rho_U^{a,b} = 0$ maximizes the determinant of P_{Q^2} can be seen as follows. Using (D.2), (D.3), (D.4), and the fact that $B_U^b = \begin{bmatrix} -L_U^b & 1 \end{bmatrix}$ we have that

$$\det(P_{Q^2}) = \det(P_{U_a^2}) \left[\varepsilon_U^b - \left(\delta_U^{a,b}\right)^2 \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} P_{U_a^2}^{-1} \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T\right] . \tag{D.10}$$

Applying (D.7) and the definition of $\rho_U^{a,b}$ (cf., (6.32)), we have

$$\det(P_{Q^2}) = \det(P_{U_a^2}) \varepsilon_U^b \left[1 - \left(\rho_U^{a,b}\right)^2\right] . \tag{D.11}$$

Hence, the choice of $\rho_U^{a,b} = 0$ maximizes the determinant. ■

■ D.2 Proof of Proposition 6.5.1

Proof. Choose $e \in E^0$. For some $a \in [0 : N-k]$, $e \in \ell \times \ell$ where $\ell \triangleq [a : a+k]$, an interval of length $k+1$. We will show that every such ℓ is a subset of $\mathcal{C}_s$ for some s. This will complete the proof because $\mathcal{C}_s$ is a clique so $\ell \times \ell \subset \mathcal{C}_s \times \mathcal{C}_s \subset E^n$. Now either ℓ contains both elements $i4k-1$ and $i4k$ (elements on a tree boundary) for some $i \in \{1, 2, \ldots, 2^M - 1\}$ or it does not. If not, then $\ell \subset \eta(s) \subset \mathcal{C}_s$ such that $m(s) = M$ and

$$\begin{aligned} \imath(s)4k &\le a \,, && \text{(D.12a)} \\ (\imath(s)+1)4k - 1 &\ge a + k \,. && \text{(D.12b)} \end{aligned}$$

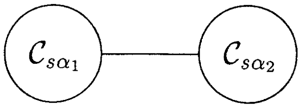


Figure D.1. The graph L^s for $m(s) = M - 1$.

Indeed, $\imath(s)4k$ and $(\imath(s) + 1)4k - 1$ are the maximum and minimum elements of $\eta(s)$ which contains an interval of length $4k$ by definition (for $m(s) = M$).

Next, consider the case that ℓ contains $i4k - 1$ and $i4k$. Then it crosses a tree boundary and belongs to some C_s for $m(s) < M$ because, by construction, the $\{C_s\}$ collectively contain all sets $[i4k - k : i4k + k - 1]$, an interval of length $2k$ around $i4k - 1$ and $i4k$. In particular, we have $\ell \subset \eta(s) \subset C_s$ for s such that

$$\imath(s)4k2^{M-m(s)} + 4k2^{M-m(s)-1} - k \leq a\,, \tag{D.13a}$$

$$\imath(s)4k2^{M-m(s)} + 4k2^{M-m(s)-1} + k - 1 \geq a + k\,. \tag{D.13b}$$

■

■ D.3 Proof of Proposition 6.5.2

To prove Proposition 6.5.2 we require several intermediate results concerning the graph L^s which is defined in Section 6.5 and illustrated in Figure 6.13. The first lemma, which follows, is clear from Figure 6.13.

Lemma D.3.1. *L^s is a tree.*

Proof. We will show this by induction. For $m(s) = M - 1$, L^s has $N_{\mathrm{v}}^{m(s)} = 2$ vertices and $N_{\mathrm{e}}^{m(s)} = 1$ edge as shown in Figure D.1. For s such that $m(s) = n$, L^s has

$$N_{\mathrm{v}}^{n} = 2N_{\mathrm{v}}^{n-1} + 2 \tag{D.14a}$$

vertices and

$$N_{\mathrm{e}}^{n} = 2N_{\mathrm{e}}^{n-1} + 3 \tag{D.14b}$$

edges, where we have used the fact that L^s has subtrees $L^{s\alpha_1}$ and $L^{s\alpha_2}$ each of which has N_{v}^{n-1} vertices and N_{e}^{n-1} edges and are joined to form L^s as shown in Figure 6.13. Using (D.14) and the inductive assumption that $N_{\mathrm{v}}^{n-1} = N_{\mathrm{e}}^{n-1} + 1$, it is straightforward to show that $N_{\mathrm{v}}^{n} = N_{\mathrm{e}}^{n} + 1$. ■

We will show that L^s is a junction tree for a specific subgraph of G^n. The fact that L^0 is a junction tree for G^n will follow immediately. We will need the following lemma.

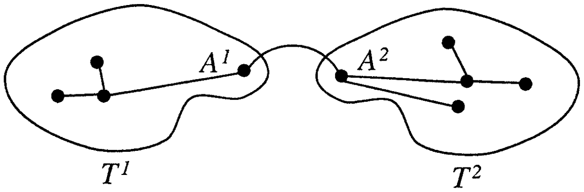


Figure D.2. The junction tree T is formed by joining two others T^1 (which has a vertex A^1) and T^2 (which has a vertex A^2). To join them, an edge is added between A^1 and A^2.

Lemma D.3.2. *Let $T^i = (\mathcal{K}^i, \mathcal{E}^i)$ be junction trees with the $\mathcal{K}^i$ disjoint. Suppose $A^i \in \mathcal{K}^i$, $i = 1,2$. Let $T = (\mathcal{K}, \mathcal{E})$ where $\mathcal{K} = \mathcal{K}^1 \cup \mathcal{K}^2$ and $\mathcal{E} = \{(A^1, A^2)\} \cup \mathcal{E}^1 \cup \mathcal{E}^2$. Then T is a junction tree if* $\text{ucli}(T^1) \cap \text{ucli}(T^2) \subseteq A^1 \cap A^2$.

The following proof follows from a straightforward application of the intersection property of junction trees (cf., Definition 6.2.6). Indeed, as shown in Figure D.2, T is comprised of T^1 and T^2 which are, themselves junction trees and have the intersection property. Junction trees T^1 and T^2 are joined by an edge between A^1 (a vertex of T^1) and A^2 (which is a vertex of T^2). The condition that $\text{ucli}(T^1) \cap \text{ucli}(T^2) \subseteq A^1 \cap A^2$ guarantees that the intersection property holds for all of T.

Proof. T is a tree because, using the fact that the $\mathcal{K}^i$ are disjoint, $|\mathcal{K}| = |\mathcal{K}^1| + |\mathcal{K}^2| = |\mathcal{E}^1| + |\mathcal{E}^2| + 2 = |\mathcal{E}| + 1$. We will show that for any $v \in \text{ucli}(T)$, $T_{\mathcal{K}_v}$ is a subtree. First, consider $v \in \text{ucli}(T^1)$ but $v \notin \text{ucli}(T^2)$. Then $\mathcal{K}_v = \mathcal{K}_v^1 \cup \mathcal{K}_v^2 = \mathcal{K}_v^1$. Hence, $T_{\mathcal{K}_v} = T_{\mathcal{K}_v^1} = T^1_{\mathcal{K}_v^1}$ is a subtree because T^1 is a junction tree. By a similar argument $T_{\mathcal{K}_v}$ is a subtree for $v \notin \text{ucli}(T^1)$ but $v \in \text{ucli}(T^2)$.

Next, consider $v \in \text{ucli}(T^1) \cap \text{ucli}(T^2)$ which implies, by assumption, that $v \in A^i$ for $i = 1,2$. Using this, and the fact that T^i is a junction tree for $i = 1,2$, it is clear that $T_{\mathcal{K}_v}$ has the intersection property. So it is a junction tree and a subtree of T. ■

Having shown that L_s is a tree in Lemma D.3.1, we will use Lemma D.3.2 to show, in the following proposition, that it is a junction tree for a particular subgraph of G^n.

Proposition D.3.1. *L^s is a junction tree for $G^n_{\text{ucli}(L^s)}$.*

Proof. It is clear that the vertex set for L^s is the set of maximal cliques of $G^n_{\text{ucli}(L^s)}$. It is also clear that L^s is a junction tree for s such that $m(s) = M - 1$ (see Figure D.1). We now proceed inductively. For a general s such that $m(s) < M - 1$, L^s has as subtrees $L^{s\alpha_i}$ for $i = 1,2$ which are themselves junction trees. To form L^s we need to join these two junction trees with the junction tree D^s which has vertices $\mathcal{C}_{s\alpha_i}$ for $i = 1,2$ and an edge $(\mathcal{C}_{s\alpha_1}, \mathcal{C}_{s\alpha_2})$. (Note that D^s is isomorphic to L^r for $m(r) = M - 1$.) That is, we need to add edges $(\mathcal{C}_{s\alpha_1}, \mathcal{C}_{s\alpha_1\alpha_2})$, $(\mathcal{C}_{s\alpha_2}, \mathcal{C}_{s\alpha_2\alpha_1})$ to join $L^{s\alpha_i}$ and D^s for $i = 1,2$. Notice that $\text{vert}(D^s)$, $\text{vert}(L^{s\alpha_1})$, and $\text{vert}(L^{s\alpha_2})$ are disjoint. Finally, note that $\text{ucli}(L^{s\alpha_1}) \cap \text{ucli}(L^{s\alpha_2}) = \emptyset$ and $\text{ucli}(D^s) \cap \text{ucli}(L^{s\alpha_i}) = \eta(s\alpha_i) \in \mathcal{C}_{s\alpha_i} \cap \mathcal{C}_{s\alpha_i\alpha_j}$ where $j = 3 - i$. Hence, by Lemma D.3.2 the proposition follows. ■

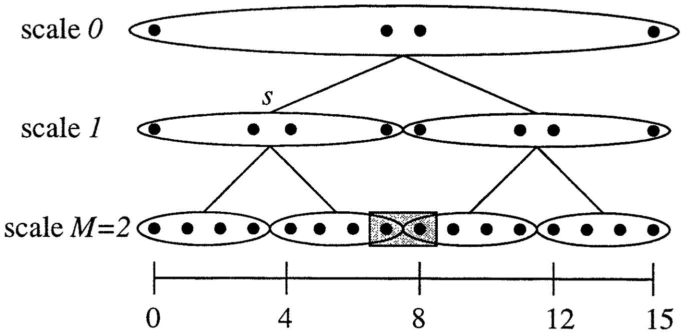


Figure D.3. An illustration of the set $\mathcal{B}_s$ (shaded) for the case $M = 2$, $k = 1$, $m(s) = 1$, $\imath(s) = 0$.

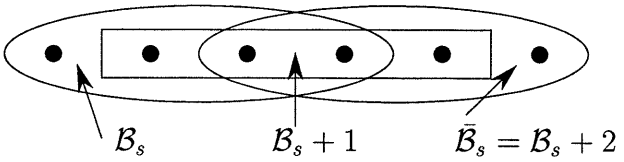


Figure D.4. An illustration of vert(F^s) for the case of $k = 3$. Each of the $k = 3$ elements of vert(F^s) is an interval of length $k + 1 = 4$ and the intervals are offset by one.

The proof of Proposition 6.5.2 follows immediately.

D.4 Proof of Proposition 6.5.3

We show that H^n, as defined in Section 6.5, is chordal by exhibiting a junction tree for it. The maximal cliques of H^n take two forms. First, recall that the maximal cliques of the original graph, G^0, are length $k + 1$ intervals. Some of these intervals are also maximal cliques of H^n while other intervals have been subsumed by sets of the form $\mathcal{C}_s$. Thus, in addition to some intervals, the maximal cliques of H^n includes a subset of $\{\mathcal{C}_s\}_{s \in \mathcal{S}_0 - \{0\}}$. Specifically, $\mathcal{C}_s$ for s such that $m(s) \geq M - n + 1$ is a maximal clique of H^n. Hence, to define a junction tree for H^n we will need to account for both types of maximal cliques, intervals and $\mathcal{C}_s$.

In this section, we will rely on the notation and definitions of Chapter 6. However, to simplify notation, we make the following definitions. Let $\mathcal{B}_s$ be the length-$(k + 1)$ interval that overlaps a tree boundary (as illustrated in Figure D.3) defined by $\mathcal{B}_s \triangleq (\imath(s) + 1)4k2^{M-m(s)} + [-k : 0]$. And, let $\bar{\mathcal{B}}_s \triangleq \mathcal{B}_s + k - 1$. Also, let F^s be the graph whose vertex set is $\text{vert}(F^s) \triangleq \{\mathcal{B}_s + j\}_{j=0}^{k-1}$ (see Figure D.4) and whose edge set is $\text{edge}(F^s) \triangleq \{(\mathcal{B}_s + j, \mathcal{B}_s + j + 1)\}_{j=0}^{k-2}\}$. Notice that the vertex set for F^s includes $\bar{\mathcal{B}}_s$ and that F^s is a tree. In fact, it is a junction tree as the following lemma verifies.

Lemma D.4.1. *F^s has the intersection property.*

The lemma follows from the fact that F^s consists of a collection of intervals offset by

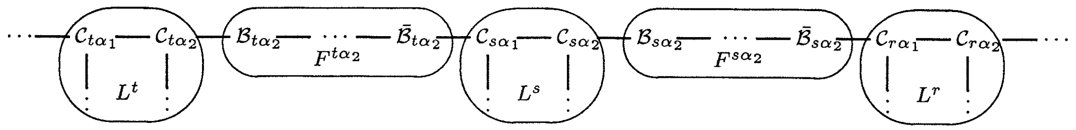

$\mathcal{C}_{t\alpha_1}$
$\mathcal{C}_{t\alpha_2}$
L^t
$\mathcal{B}_{t\alpha_2}$
$\bar{\mathcal{B}}_{t\alpha_2}$
$F^{t\alpha_2}$
$\mathcal{C}_{s\alpha_1}$
$\mathcal{C}_{s\alpha_2}$
L^s
$\mathcal{B}_{s\alpha_2}$
$\bar{\mathcal{B}}_{s\alpha_2}$
$F^{s\alpha_2}$
$\mathcal{C}_{r\alpha_1}$
$\mathcal{C}_{r\alpha_2}$
L^r

Figure D.5. The graph T^n where $m(r) = m(s) = m(t) = M - n$ and $i(t) = \imath(s) - 1 = i(r) - 2$.

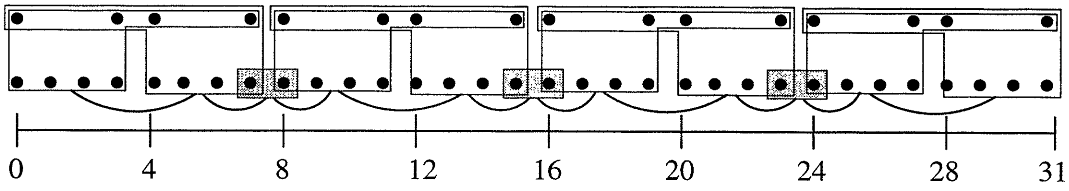


Figure D.6. The enclosed elements represent the vertices of T^1 for the case $k = 1$, $M = 3$. Included among these are the shaded boxes which represent the sets (from left to right) $\mathcal{B}_{(M,1)}$, $\mathcal{B}_{(M,3)}$, and $\mathcal{B}_{(M,5)}$. The other boxes represent the $\mathcal{C}_{(M,i)}$. Also shown are the edges of T^1.

one as shown in Figure D.4. An algebraic proof follows.

Proof. Let A_i, for $i = 1, 2, 3$ be vertices of F^s where, A_2 is on the (unique) length-n path from A_1 to A_3. Let $a^*(i)$ $(a_*(i))$ be the maximal (minimal) element of A_i. Assume, without loss of generality, that the maximal element of $a^*(1) < a^*(3)$. Of course, we must have $a^*(1) \leq a^*(2) \leq a^*(3)$ and $a_*(1) \leq a_*(2) \leq a_*(3)$. By definition of F^s, $A_1 \cap A_3 = [a_*(1) + n : a^*(3) - n]$. Suppose the path from A_1 to A_2 is length $\ell \leq n$. Then $A_2 \ni a_*(1) + \ell \leq a_*(1) + n$. Suppose that the path from A_2 to A_3 is length $p \leq n$ Then $A_2 \ni a^*(3) - p \geq a^*(3) - n$. The lemma follows because $A_2 \supseteq [a_*(1) + \ell : a^*(3) - p] \supseteq [a_*(1) + n : a^*(3) - n]$. ■

To define a junction tree for H^n, we combine a collection of graphs $\{L^s\}$ (the graph L^s is defined in Section 6.5 and discussed in Section D.3) with a collection of graphs $\{F^s\}$. Each graph L^s is a dyadic tree corresponding to a subset of the collection of maximal cliques $\{\mathcal{C}_s\}$ (see Figure 6.13). The collection $\{L^s\}$ represent some of the maximal cliques of H^n, namely $\{\mathcal{C}_s\}_{s:m(s)\geq M-n+1}$. The remaining maximal cliques of H^n are intervals and these are represented by $\{F^s\}$. We will join together several dyadic trees, L^s, by linking pairs of them with graphs of the form F^s. This is illustrated abstractly in Figure D.5 and for a concrete example in Figure D.6 (to be described in greater detail shortly). Specifically, we define T^n to be the graph whose vertex set is the disjoint union

$$\mathrm{vert}(T^n) \triangleq \bigsqcup_{s\in\mathcal{T}_0(M-n)} \left(\mathrm{vert}(L^s) \sqcup \mathrm{vert}(F^{s\alpha_2})\right) \tag{D.15}$$

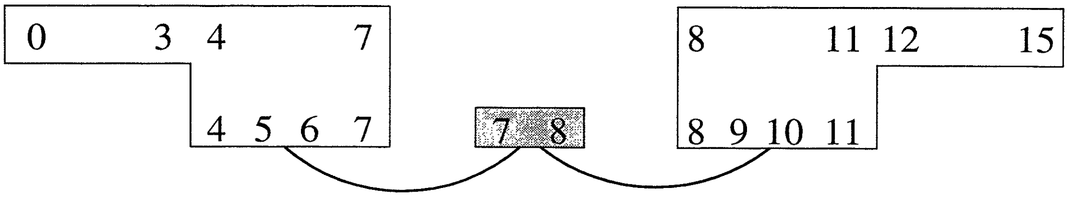


Figure D.7. A subgraph of the graph depicted in Figure D.6.

and whose edge set is

$$\text{edge}(T^n) = \bigcup_{s \in \mathcal{T}_0(M-n)} \Big(\text{edge}(L^s) \cup \text{edge}(F^{s\alpha_2}) \cup \{(\mathcal{C}_{s\alpha_2}, \mathcal{B}_{s\alpha_2})\} \cup \{(\mathcal{C}_{s\alpha_1}, \bar{\mathcal{B}}_{s\alpha_1})\} \Big) \tag{D.16}$$

The graph T^n is depicted in Figure D.5. Notice that T^n contains as subgraphs L^s and $F^{s\alpha_2}$ for all $s \in \mathcal{T}_0(M-n)$ as indicated in Figure D.5. Figure D.6 illustrates T^1 for the case in which $k = 1$, $M = 3$. The shaded boxes represent the sets (from left to right) $\mathcal{B}_{(M,1)}$, $\mathcal{B}_{(M,3)}$, and $\mathcal{B}_{(M,5)}$ where we are using the notation $s = (m(s), \imath(s))$. The other boxes represent the $\mathcal{C}_{(M,i)}$. The arcs linking boxes are the edges of T^1. Figure D.6 is a bit cluttered and the structure of T^1 is hard to see. However, by focusing on a subgraph of T^1, its structure is revealed a bit more clearly (see Figure D.7).

Proposition D.4.1. *T^n is a junction tree for H^n.*

The proof follows, more or less, by construction and, intuitively, it is clear from Figure D.5 and Figure D.6 that T^n is a junction tree for H^n. It has already been shown that each graph L^s and F^s is a tree. Since T^n is just a linking of such graphs, a proof follows from Lemma D.3.2.

Proof. T^n is a tree. Indeed, a careful count reveals that the number of vertices of T^n is one more than the number of edges. By an argument similar to the one made in the proof of Proposition 6.5.1 it can be verified that $\text{vert}(T^n)$ are the maximal cliques of H^n. It is easy to check that $\text{ucli}(L^s) \cap \text{ucli}(F^{s\alpha_2}) \subset \mathcal{C}_{s\alpha_2} \cap \mathcal{B}_{s\alpha_2}$ and $\text{ucli}(L^s) \cap \text{ucli}(F^{t\alpha_2}) \subset \mathcal{C}_{s\alpha_1} \cap \bar{\mathcal{B}}_{t\alpha_2}$ for $\imath(t) = \imath(s) - 1$. We also have that $\text{ucli}(L^t) \cap \text{ucli}(L^s) = \text{ucli}(F^{t\alpha_2}) \cap \text{ucli}(F^{s\alpha_2}) = \text{ucli}(L^s) \cap \text{ucli}(F^{r\alpha_2}) = \emptyset$ for $t \neq s$ and for all r such that $\imath(r) \neq \imath(s) \pm 1$. Therefore, by Lemma D.3.2 the proposition follows. ■

That H^n is chordal follows immediately from Proposition 6.2.2 and Proposition D.4.1.

■ D.5 Sketch of a Proof of Proposition 6.5.4

In this section we sketch a proof of Proposition 6.5.4 pictorially for the case of $M = 3$ and $k = 1$. Specifically, we will illustrate a sequence of junction trees corresponding to a sequence of graphs beginning with H^{n-1} and ending with H^n. Moreover, this sequence

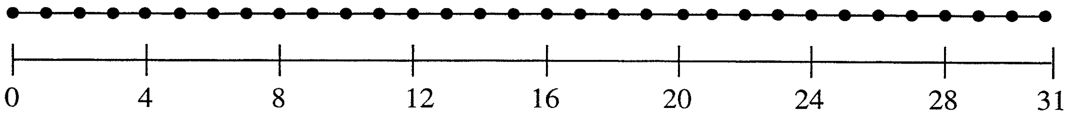


Figure D.8. The graph $H^0 \triangleq G^0$ for $M = 3$, $k = 1$.

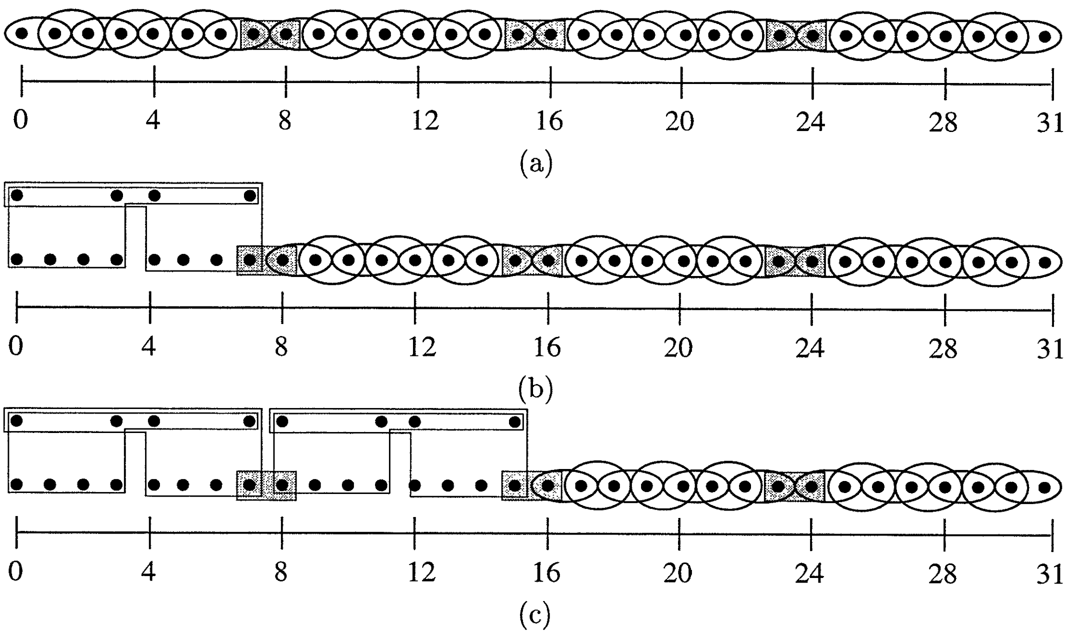


Figure D.9. (a)–(c) The vertices for junction trees in the sequence of junction trees for graphs starting with G^0 and ending with H^1 for the case $M = 3$, $k = 1$.

of graphs will have the property described in Proposition 6.5.4, i.e., that for all s and t such that $m(s) = m(t) = M - n + 1$ and $\imath(s) < \imath(t)$, the edges of $\mathcal{C}_s$ are added prior to those of $\mathcal{C}_t$. Our starting point is graph $H^0 \triangleq G^0$ for the case $M = 3$, $k = 1$ which is illustrated in Figure D.8. The maximal cliques of H^0 (i.e., the vertices of the junction tree for H^0) consist of length-$k+1$ intervals and are illustrated in Figure D.9(a). The boxed and shaded cliques of Figure D.9(a) are those that will remain in H^1 as shown in Figure D.6. Each of the others will be subsumed by a set $\mathcal{C}_s$.

Consider adding edges to form maximal cliques $\mathcal{C}_{s_1}$ and $\mathcal{C}_{s_2}$ for s_1 and s_2 the two left-most leaf nodes (i.e., $m(s_1) = m(s_2) = M$ and $\imath(s_1) = \imath(s_2) - 1 = 0$). The vertices of the resulting junction tree are shown in Figure D.9(b). Notice that some of the maximal cliques representing intervals illustrated in Figure D.9(a) have been subsumed by $\mathcal{C}_{s_1}$ and $\mathcal{C}_{s_2}$. Now consider adding the edges to form the maximal cliques $\mathcal{C}_{t_1}$ and $\mathcal{C}_{t_2}$ where t_1 and t_2 are the left-most leaf nodes just to the right of s_2 (i.e., $m(t_1) = m(t_2) = M$ and $\imath(t_1) = \imath(t_2) - 1 = 2$). The vertices of the resulting junction tree are shown in Figure D.9(c). Again, some of the interval cliques have been subsumed. Continuing this process of adding the edges for C_s according to increasing $\imath(s)$, we arrive at the junction tree for H^1 whose vertices are illustrated in Figure D.6. Three of the original interval cliques remain.

A similar procedure—adding the C_s in order of increasing $\imath(s)$—can be applied scale-recursively, beginning at the next coarser scale $(M - 1)$ and continuing until scale 1. Doing so will yield a sequence of graphs with the properties described in Proposition 6.5.4. Although not explicitly shown here, this procedure can be applied for *any* M and k. This completes our sketch of a proof of Proposition 6.5.4. Formalizing this sketch requires carefully defining the sequence of junction trees we have illustrated (for a special case). However, doing so will not provide any insight and the formality will, no doubt, obscure the simplicity of the procedure.

Appendix E

Proofs for Chapter 7

In this appendix we provide proofs for Proposition 7.3.1 (Section E.1) and Proposition 7.3.2 (Section E.2).

■ E.1 Proof of Proposition 7.3.1

Proof. Let us proceed by induction on $n \in \{m - m(r) - 1, m - m(r) - 2, \dots, 0\}$. First note, that $P_{x(r)} = P_{f(r)} = L_r P_{fM} L_r^T$.

Base Case: $n = m - m(r) - 1$: For this case $t\bar{\gamma}^{n+1} = s\bar{\gamma}^{n+1} = r$. Let $t_1 = t\bar{\gamma}^n$ and $s_1 = s\bar{\gamma}^n$. Then,

$$\mathrm{E}[x(t_1)x(r)^T] = \mathrm{E}[(A(t_1)x(r) + w(t_1))x(r)^T] \tag{E.1a}$$

$$= A(t_1)E[x(r)x(r)^T] \tag{E.1b}$$

$$= L_{t_1} P_{fM} L_r^T (L_r P_{fM} L_r^T)^{-1} L_r P_{fM} L_r^T \tag{E.1c}$$

$$= L_{t_1} P_{fM} L_r^T\,. \tag{E.1d}$$

We also have,

$$\mathrm{E}[x(t_1)x(s_1)^T] = \mathrm{E}[x(t_1)(A(s_1)x(r) + w(s_1))^T] \tag{E.2a}$$

$$= \mathrm{E}[x(t_1)x(r)^T]A(s_1)^T \tag{E.2b}$$

$$= L_{t_1} P_{fM} L_r^T (L_r P_{fM} L_r^T)^{-1} L_r P_{fM} L_{s_1}^T \tag{E.2c}$$

$$= L_{t_1} P_{fM} L_{s_1}^T \tag{E.2d}$$

where we have used (E.1) and (7.19a) (or, equivalently, (7.19b)) with Lemma 7.3.1.

Inductive Assumption: Assume that for some $n + 1 \in \{m - m(r) - 1, m - m(r) - 2, \dots, 0\}$, $\mathrm{E}[x(t\bar{\gamma}^{n+1})x(s\bar{\gamma}^{n+1})^T] = L_{t\bar{\gamma}^{n+1}} P_{fM} L_{s\bar{\gamma}^{n+1}}^T$.

Inductive Step: We have

$$\begin{aligned}
\mathrm{E}[x(t\bar{\gamma}^n)x(s\bar{\gamma}^n)^T] &= A(t\bar{\gamma}^n)\,\mathrm{E}[x(t\bar{\gamma}^{n+1})x(s\bar{\gamma}^{n+1})^T]A(s\bar{\gamma}^n)^T && \text{(E.3a)}\\
&= L_{t\bar{\gamma}^n}P_{fM}L^T_{t\bar{\gamma}^{n+1}}(L_{t\bar{\gamma}^{n+1}}P_{fM}L^T_{t\bar{\gamma}^{n+1}})^{-1}L_{t\bar{\gamma}^{n+1}}P_{fM}L^T_{s\bar{\gamma}^{n+1}}\\
&\qquad\times(L_{s\bar{\gamma}^{n+1}}P_{fM}L^T_{s\bar{\gamma}^{n+1}})^{-1}L_{s\bar{\gamma}^{n+1}}P_{fM}L^T_{s\bar{\gamma}^n} && \text{(E.3b)}\\
&= L_{t\bar{\gamma}^n}P_{fM}L^T_{s\bar{\gamma}^{n+1}}(L_{s\bar{\gamma}^{n+1}}P_{fM}L^T_{s\bar{\gamma}^{n+1}})^{-1}L_{s\bar{\gamma}^{n+1}}P_{fM}L^T_{s\bar{\gamma}^n} && \text{(E.3c)}\\
&= L_{t\bar{\gamma}^n}P_{fM}L^T_{s\bar{\gamma}^n} && \text{(E.3d)}
\end{aligned}$$

where in (E.3a) we've used the MAR dynamics, in (E.3b) we've used the inductive assumption, in (E.3c) we've used (7.19a) and Lemma 7.3.1, and in (E.3d) we've used (7.19b) and Lemma 7.3.1. This completes the proof. ■

■ E.2 Proof of Proposition 7.3.2

Proof. We will proceed by induction on $j \in \{m(t)-m(s),\ldots,0\}$.

Base Case: $j = m(t)-m(s)$: In this case $t\bar{\gamma}^j = t\bar{\gamma}^{m(t)-m(s)} = t'$ by definition. Note that $m(t') = m(s)$. Then, by Proposition 7.3.1 and (7.20a) and (7.20b) we have

$$\mathrm{E}[x(t')x(s)^T] = L_{t'}P_{fM}L_s^T\,. \tag{E.4}$$

Inductive Assumption: Assume that for some $j+1 \in \{m(t)-m(s), m(t)-m(s)-2,\ldots,0\}$ we have that

$$\mathrm{E}[x(t\bar{\gamma}^{j+1})x(s)^T] = L_{t\bar{\gamma}^{j+1}}P_{fM}L_s^T\,. \tag{E.5}$$

Inductive Step: We have

$$\begin{aligned}
\mathrm{E}[x(t\bar{\gamma}^j)x(s)^T] &= A(t\bar{\gamma}^j)\,\mathrm{E}[x(t\bar{\gamma}^{j+1})x(s)^T] && \text{(E.6a)}\\
&= L_{t\bar{\gamma}^j}P_{fM}L^T_{t\bar{\gamma}^{j+1}}(L_{t\bar{\gamma}^{j+1}}P_{fM}L^T_{t\bar{\gamma}^{j+1}})^{-1}\,\mathrm{E}[x(t\bar{\gamma}^{j+1})x(s)^T] && \text{(E.6b)}\\
&= L_{t\bar{\gamma}^j}P_{fM}L^T_{t\bar{\gamma}^{j+1}}(L_{t\bar{\gamma}^{j+1}}P_{fM}L^T_{t\bar{\gamma}^{j+1}})^{-1}L_{t\bar{\gamma}^{j+1}}P_{fM}L_s^T && \text{(E.6c)}\\
&= L_{t\bar{\gamma}^j}P_{fM}L_s^T && \text{(E.6d)}
\end{aligned}$$

where we have used the inductive assumption, (7.21) and Lemma 7.3.1. This completes the proof. ■

Bibliography

[1] K. Abend, T. Harley, and L. Kanal. Classification of binary random patterns. *IEEE Transactions on Information Theory*, 11(4):538–544, October 1965.

[2] Y. Abramovich, N. Spencer, and A. Gorokhov. Positive-definite Toeplitz completion in DOA estimation for nonuniform linear antenna arrays—part II: partially augmentable arrays. *IEEE Transactions on Signal Processing*, 47(6):1502–1521, June 1999.

[3] P. Abry, P. Goncalves, and P. Flandrin. *Wavelets and Statistics*, chapter Wavelets, spectrum analysis, and $1/f$ processes. Springer-Verlag, New York, 1995.

[4] P. Abry and F. Sellan. The wavelet-based synthesis for fractional Brownian motion proposed by F. Sellan and Y. Meyer: Remarks and fast implementation. *Applied and Computational Harmonic Analysis*, 3:377–383, 1996.

[5] M. Adams, A. Willsky, and B. Levy. Linear estimation of boundary value stochastic processes—part II: 1-d smoothing problems. *IEEE Transactions on Automatic Control*, 29(9):811–821, September 1984.

[6] H. Akaike. Stochastic theory of minimal realizations. *IEEE Transactions on Automatic Control*, AC–19(6):667–674, December 1974.

[7] H. Akaike. Markovian representation of stochastic processes by canonical variables. *SIAM Journal of Control*, 13(1):162–173, January 1975.

[8] K. Arun and S. Kung. Balanced approximation of stochastic systems. *SIAM Journal of Matrix Analysis and Applications*, 11(1):42–68, January 1990.

[9] M. Barnsley, R. Devaney, B. Mandelbrot, H. Peitgen, D. Saupe, and R. Voss. *The Science of Fractal Images.* Springer-Verlag, 1988.

[10] W. Barrett, C. Johnson, and R. Loewy. Critical graphs for the positive definite completion problem. *SIAM Journal on Matrix Analysis and Applications*, 20(1):117–130, 1999.

[11] W. Barrett, C. Johnson, and M. Lundquist. Determinantal formulae for matrix completions associated with chordal graphs. *Linear Algebra and its Applications*, 121:265–289, 1989.

[12] M. Basseville, A. Benveniste, K. Chou, S. Golden, R. Nikoukhah, and A. Willsky. Modeling and estimation of multiresolution stochastic processes. *IEEE Transactions on Information Theory*, 38(2):766–784, March 1992.

[13] M. Basseville, A. Benveniste, and A. Willsky. Multiscale autoregressive processes, part I: Schur-Levinson parameterizations. *IEEE Transactions on Signal Processing*, 40(8):1915–1934, August 1992.

[14] M. Basseville, A. Benveniste, and A. Willsky. Multiscale autoregressive processes, part II: lattice structures for whitening and modeling. *IEEE Transactions on Signal Processing*, 40(8):1935–1944, August 1992.

[15] J. Beekman and E. Shiu. Stochastic models for bond prices, function space integrals, and immunization theory. *Insurance Mathematics and Economics*, 7(3):163–173, October 1988.

[16] A. Benveniste, R. Nikoukhah, and A. Willsky. Multiscale system theory. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 41(1):2–14, January 1994.

[17] J. Beran. *Statistics for Long-Memory Processes.* Chapman and Hall, New York, 1994.

[18] C. Berge. *Graphs and Hypergraphs.* North-Holland, New York, 1976.

[19] N. Biggs, E. Loyd, and R. Wilson. *Graph Theory 1736–1936.* Clarendon Press, Oxford, England, 1976.

[20] W. Briggs. *A Multigrid Tutorial.* SIAM, 1987.

[21] P. Brockwell and R. Davis. *Time Series: Theory and Methods.* Springer-Verlag, New York, 1987.

[22] J. Burg. Maximum entropy spectral analysis. In *Proceedings of the 37th Annual International Meeting of The Society of Exploratory Geophysics*, Oklahoma City, OK, October 1967.

[23] J. Burg. *Maximum entropy spectral analysis.* PhD thesis, Stanford University, Stanford, CA, 1975.

[24] P. Burt and E. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM–31(4):532–540, April 1983.

[25] B. Chen, C. Lin, and Y. Chen. Optimal signal reconstruction in noisy filter bank systems: multirate Kalman synthesis filtering approach. *IEEE Transactions on Signal Processing*, 43(11):2496–2504, November 1995.

[26] H. Cheng and C. Bouman. Trainable context model for multiscale segmentation. In *Proceedings of the IEEE International Conference on Image Processing*, Chicago, IL, October 1998.

[27] Y. Cheng and B. Chen. Nonuniform filter bank design with noises. *IEEE Transactions on Signal Processing*, 46(9):2326–2344, September 1998.

[28] K. Chou. *A stochastic modeling approach to multiscale signal processing.* PhD thesis, Massachusetts Institute of Technology, May 1991.

[29] K. Chou, A. Willsky, and A. Benveniste. Multiscale recursive estimation, data fusion, and regularization. *IEEE Transactions on Automatic Control*, 39(3):464–478, March 1994.

[30] K. Chou, A. Willsky, and R. Nikoukhah. Multiscale systems, Kalman filters, and Riccati equations. *IEEE Transactions on Automatic Control*, 39(3):479–492, March 1994.

[31] C. Chui. *Wavelets: A Tutorial in Theory and Applications.* Academic Press, 1992.

[32] A. Cohen, I. Daubechies, and J. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45:485–560, 1992.

[33] T. Constantinescu. *Schur Parameters, Factorization, and Dilation Problems.* Birkhauser Verlag, Basel, 1991.

[34] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley and Sons, New York, 1991.

[35] M. Crouse and R. Baraniuk. Simplified wavelet-domain hidden Markov models using contexts. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2277–2280, Seattle, WA, May 1998.

[36] M. Crouse, R. Nowak, and R. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4):886–902, April 1998.

[37] M. Daniel. *Multiresolution statistical modeling with application to modeling groundwater flow.* PhD thesis, Massachusetts Institute of Technology, February 1997.

[38] M. Daniel and A. Willsky. Modeling and estimation of fractional Brownian motion using multiresolution stochastic processes. In J. L. Vehel, E. Lutton, and C. Tricot, editors, *Fractals in Engineering*, pages 124–137. Springer, 1997.

[39] M. Daniel and A. Willsky. A multiresolution methodology for signal-level fusion and data assimilation with applications to remote sensing. *Proceedings of the IEEE*, 85(1):164–180, January 1997.

[40] M. Daniel and A. Willsky. The modeling and estimation of statistically self-similar processes in a multiresolution framework. *IEEE Transactions on Information Theory*, 45(3):955–970, April 1999.

[41] M. Daniel, A. Willsky, and D. McLaughlin. Travel time estimation using a multiscale stochastic framework. *Advanced Water Resources.* Submitted.

[42] K. Daoudi, April 1999. Private communication.

[43] K. Daoudi, A. Frakt, and A. Willsky. Multiscale autoregressive models and wavelets: extended version. Technical Report LIDS-P-2437, Massachusetts Institute of Technology, November 1998.

[44] K. Daoudi, A. Frakt, and A. Willsky. Multiscale autoregressive models and wavelets. *IEEE Transactions on Information Theory*, 45(3):828–845, April 1999.

[45] J. Darroch, S. Lauritzen, and T. Speed. Markov fields and log-linear interaction models for contingency tables. *The Annals of Statistics*, 8(3):522–539, 1980.

[46] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, November 1988.

[47] I. Daubechies. The wavelet transform, time-frequency localization, and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, September 1990.

[48] I. Daubechies. *Ten Lectures on Wavelets.* SIAM, 1992.

[49] B. De Moor, P. Van Overschee, and J. Suykens. Subspace algorithms for system identification and stochastic realization. In *Proceedings of the International Symposium on Recent Advances in Mathematical Theory of Systems, Control, Networks, and Signal Processing (MTNS '91)*, pages 589–595, Kobe, Japan, June 1991.

[50] A. Delopoulos and S. Kollias. Optimal filter banks for signal reconstruction from noisy subband components. *IEEE Transactions on Signal Processing*, 44(2):212–224, February 1996.

[51] A. Dempster. Covariance selection. *Biometrics*, 28:157–175, March 1972.

[52] J. Dennis and K. Turner. Generalized conjugate directions. *Linear Algebra and its Applications*, 88/89:187–209, 1987.

[53] H. Derin and P. Kelly. Discrete-index Markov-type random processes. *Proceedings of the IEEE*, 77(10):1485–1510, October 1989.

[54] U. Desai and D. Pal. A realization approach to stochastic model reduction and balanced stochastic realizations. In *IEEE Conference on Decision and Control*, pages 1105–1112, 1982.

[55] P. Dewilde and E. Deprettere. Modeling VLSI interconnections as an inverse scattering problem. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 147–153, 1987.

[56] C. Dietrich and G. Newsam. Fast and exact simulation of stationary Gaussian processes through the circulant embedding of the covariance matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107, July 1997.

[57] I. Duff, A. Erisman, and J. Reid. *Direct Methods for Sparse Matrices.* Clarendon Press, 1986.

[58] H. Dym and I. Gohberg. Extensions of band matrices with band inverses. *Linear Algebra and its Applications*, 36:1–24, 1981.

[59] R. Ellis, I. Gohberg, and D. Lay. Band extensions, maximum entropy and the permanence principle. In J. Justice, editor, *Maximum Entropy and Bayesian Methods in Applied Statistics*, pages 131–155. Cambride University Press, New, 1986.

[60] D. Entekhabi, H. Nakamura, and E. Njoku. Solving the inverse problem for soil moisture and temperature profiles by sequential assimilation of multifrequency remotely sensed observations. *IEEE Transactions on Geoscience and Remote Sensing*, 32(2):438–448, March 1994.

[61] A. Erisman and W. Tinney. On computing certain elements of the inverse of a sparse matrix. *Communications of the ACM*, 18(3), 1975.

[62] E. Fabre. New fast smoothers for multiscale systems. *IEEE Transactions on Signal Processing*, 44(8):1893–1911, August 1996.

[63] P. Fieguth. *Application of multiscale estimation to large scale multidimensional imaging and remote sensing problems.* PhD thesis, Massachusetts Institute of Technology, June 1995.

[64] P. Fieguth. Global-scale three-dimensional statistical estimation. In *Proceedings of the IEEE Conference on Multidimensional Signal Processing*, pages 247–250, Alpbach, Austria, 1998.

[65] P. Fieguth. Multipole-motivated reduced-state estimation. In *Proceedings of the IEEE International Conference on Image Processing*, Chicago, 1998.

[66] P. Fieguth, W. Karl, A. Willsky, and C. Wunsch. Multiresolution optimal interpolation and statistical analysis of TOPEX/POSEIDON satellite altimetry. *IEEE Transactions on Geoscience and Remote Sensing*, 33(2):280–292, March 1995.

[67] P. Fieguth, D. Menemenlis, T. Ho, A. Willsky, and C. Wunsch. Mapping Mediterranean altimeter data with a multiresolution optimal interpolation algorithm. *Journal of Atmospheric and Oceanic Technology*, 15:535–546, April 1998.

[68] P. Fieguth and A. Willsky. Fractal estimation using models on multiscale trees. *IEEE Transactions on Signal Processing*, 44(5):1297–1300, May 1996.

[69] P. Fieguth, A. Willsky, and W. Karl. Efficient multiresolution counterparts to variational methods for surface reconstruction. *Computer Vision and Image Understanding*, 70(2):157–176, May 1998.

[70] P. Flandrin. On the spectrum of fractional Brownian motions. *IEEE Transactions on Information Theory*, 35(1):197–199, January 1989.

[71] P. Flandrin. Time-scale analysis and self-similar stochastic processes. In *Proceedings of the NATO Advanced Study Institute on Wavelets and Their Applications*, Il Ciocco, Italy, August 1992.

[72] P. Flandrin. Wavelet analysis and synthesis of fractional Brownian motion. *IEEE Transactions on Information Theory*, 38(2):910–917, March 1992.

[73] C. Fosgate. Multiscale segmentation and anomaly enhancement of SAR imagery. Master's thesis, Massachusetts Institute of Technology, June 1996.

[74] C. Fosgate, H. Krim, W. Irving, and A. Willsky. Multiscale segmentation and anomaly enhancement of SAR imagery. *IEEE Transactions on Image Processing*, 6(1):7–20, January 1997.

[75] A. Frakt. Multiscale hypothesis testing with application to anomaly characterization from tomographic projections. Master's thesis, Massachusetts Institute of Technology, May 1996.

[76] A. Frakt, W. Karl, and A. Willsky. A multiscale hypothesis testing approach to anomaly detection and localization from noisy tomographic data. *IEEE Transactions on Image Processing*, 7(6):825–837, June 1998.

[77] A. Frakt and A. Willsky. Computationally efficient stochastic realization for internal multiscale autoregressive models. *Multidimensional Systems and Signal Processing*. Submitted.

[78] A. Frakt and A. Willsky. Efficient multiscale stochastic realization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, WA, May 1998.

[79] A. Frakt and A. Willsky. Multiscale autoregressive models and the stochastic realization problem. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, Asilomar, CA, November 1998.

[80] P. Frakt. *A cross-national analysis of the effects of domestic development on government responsiveness: the case of international labor standards.* PhD thesis, Douglass College, New Brunswick, NJ, 1974.

[81] B. Friedlander. Lattice methods for spectral estimation. *Proceedings of the IEEE*, 70, September 1982.

[82] A. George and J. Liu. *Computer Solution of Large and Sparse Positive Definite Systems.* Prentice Hall, Englewood Cliffs, NJ, 1981.

[83] B. Gidas. A renormalization group approach to image processing problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):164–180, February 1989.

[84] G. Golub and C. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, London, third edition, 1996.

[85] M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[86] J. Grandell, M. Hamrud, and P. Toll. A remark on the correspondence between the maximum-entropy method and the autoregressive model. *IEEE Transactions on Information Theory*, 26:750–751, November 1980.

[87] D. Griffiths. *Introduction to Electrodynamics.* Prentice Hall, Englewood Cliffs, NJ, 1989.

[88] R. Grone, C. Johnson, E. Sa, and H. Wolkowicz. Positive definite completions of partial Hermitian matrices. *Linear Algebra and its Applications*, 58:109–124, 1984.

[89] M. Hayes. *Statistical Digital Signal Processing and Modeling.* John Wiley and Sons, 1996.

[90] T. Ho. *Multiscale modeling and estimation of large-scale dynamic systems.* PhD thesis, Massachusetts Institute of Technology, September 1998.

[91] T. Ho, P. Fieguth, and A. Willsky. Multiresolution stochastic models for the efficient solution of large-scale space-time estimation problems. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3097–3100, Atlanta, GA, May 1996.

[92] T. Ho, A. Frakt, and A. Willsky. Multiscale realization and estimation for space and space-time problems. In *Proceedings of the IEEE International Symposium on Information Theory*, Cambridge, MA, August 1998.

[93] R. Horn and C. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991.

[94] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

[95] S. Ihara. Maximum entropy spectral analysis and ARMA processes. *IEEE Transactions on Information Theory*, 30:377–380, March 1984.

[96] W. Irving. *Multiscale stochastic realization and model identification with applications to large-scale estimation problems.* PhD thesis, Massachusetts Institute of Technology, September 1995.

[97] W. Irving, P. Fieguth, and A. Willsky. An overlapping tree approach to multiscale stochastic modeling and estimation. *IEEE Transactions on Image Processing*, 6(11), November 1997.

[98] W. Irving, W. Karl, and A. Willsky. A theory for multiscale stochastic realization. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, volume 1, pages 655–62, Lake Buena Vista, FL, December 1994.

[99] W. Irving, L. Novak, and A. Willsky. A multiresolution approach to discriminating targets from clutter in SAR imagery. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1157–1169, October 1997.

[100] W. Irving and A. Willsky. A canonical correlations approach to multiscale stochastic realization. *IEEE Transactions on Automatic Control.* Submitted. Preprint available at `http://vougeot.mit.edu/ssg.cgi/pubs/pubs.mpl`.

[101] E. Jaynes. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70:939–952, September 1982.

[102] C. Johnson and L. Rodman. Chordal inheritance principles and positive definite completions of partial matrices over function rings. *Operator Theory: Advances and Applications*, 35:107–127, 1988.

[103] A. Journel and C. Huijbregts. *Mining Geostatistics.* Academic Press, New York, 1978.

[104] R. Kalman. A new approach to linear filtering and prediction problems. *The American Society of Mechanical Engineers: Basic Engineering, series D*, 82:35–45, March 1960.

[105] R. Kalman and R. Bucy. New results in linear filtering and prediction theory. *The American Society of Mechanical Engineers: Basic Engineering, series D*, 83:95–108, March 1961.

[106] A. Kannan. *Adaptation of spectral trajectory models for LVCSR.* PhD thesis, Boston University, 1997.

[107] A. Kannan and S. Khudanpur. Tree-structured models of parameter dependence for rapid adaptation in large vocabulary conversational speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, 1999.

[108] A. Kannan and M. Ostendorf. Modeling dependence in adaptation of acoustic models using multiscale tree processes. In *Proceedings of EUROSPEECH*, pages 1863–1866, 1997.

[109] M. Keshner. $1/f$ noise. *Proceedings of the IEEE*, 70(3):212–218, March 1982.

[110] A. Kim. Hierarchical stochastic modeling for segmentation and compression of SAR imagery. Master's thesis, Massachusetts Institute of Technology, June 1997.

[111] A. Kim and H. Krim. Hierarchical stochastic modeling of SAR imagery for segmentation/compression. *IEEE Transactions on Signal Processing*, 47(2):458–468, February 1999.

[112] B. Kosko. *Neural Networks for Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1992.

[113] H. Krim and J. Pesquet. Multiresolution analysis of a class of nonstationary processes. *IEEE Transactions on Information Theory*, 41(4):1010–1020, July 1995.

[114] P. Kumar. A multiple scale state-space model for characterizing subgrid scale variability of near-surface soil moisture. *IEEE Transactions on Geoscience and Remote Sensing*, 37(1):182–197, January 1999.

[115] S. Lauritzen. *Graphical Models.* Oxford University Press, 1996.

[116] C. Lawson and R. Hanson. *Solving Least Squares Problems.* SIAM, Philadelphia, 1995.

[117] H. Lev-Ari. *Nonstationary Lattice-Filter Modeling.* PhD thesis, Stanford University, Stanford, CA, December 1983.

[118] H. Lev-Ari and T. Kailath. Schur and Levinson algorithms for nonstationary processes. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 860–864, Atlanta, 1981.

[119] H. Lev-Ari and T. Kailath. Lattice filter parameterization and modeling of nonstationary processes. *IEEE Transactions on Information Theory*, 30(1):1984, January 1984.

[120] H. Lev-Ari, S. Parker, and T. Kailath. Multidimensional maximum-entropy covariance extension. *IEEE Transactions on Information Theory*, 35(3):497–508, May 1989.

[121] B. Levy. Noncausal estimation for discrete Gauss-Markov random fields. In *Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems*, volume 2, pages 13–21, 1989.

[122] B. Levy, M. Adams, and A. Willsky. Solution and linear estimation of 2-d nearest-neighbor models. *Proceedings of the IEEE*, 78(4):627–641, April 1990.

[123] A. Lindquist and G. Picci. On the stochastic realization problem. *SIAM Journal on Control and Optimization*, 17(3):365–389, May 1979.

[124] S. Lovejoy and B. Mandelbrot. Fractal properties of rain, and a fractal model. *Tellus*, 37A(3):209–232, 1985.

[125] S. Lovejoy and D. Schertzer. Generalized scale invariance in the atmosphere and fractal models of rain. *Water Resources Research*, 21(8):1233–1250, August 1985.

[126] D. Luenberger. *Optimization by Vector Space Methods.* John Wiley and Sons, New York, 1969.

[127] M. Luettgen. *Image processing with multiscale stochastic models.* PhD thesis, Massachusetts Institute of Technology, May 1993.

[128] M. Luettgen, W. Karl, and A. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Transactions on Image Processing*, 3(1):41–64, January 1994.

[129] M. Luettgen, W. Karl, A. Willsky, and R. Tenney. Multiscale representations of Markov random fields. *IEEE Transactions on Signal Processing*, 41(12):3377–3396, December 1993.

[130] M. Luettgen and A. Willsky. Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination. *IEEE Transactions on Image Processing*, 4(2):194–207, February 1995.

[131] M. Luettgen and A. Willsky. Multiscale smoothing error models. *IEEE Transactions on Automatic Control*, 40(1):173–175, January 1995.

[132] M. Lundquist and C. Johnson. Linearly constrained positive definite completions. *Linear Algebra and Its Applications*, 150:195–208, 1991.

[133] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[134] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1998.

[135] B. Mandelbrot and J. Van Ness. Fractional Brownian motions, fractional noises, and applications. *SIAM Review*, 10:422–437, 1968.

[136] S. Martfnez and D. Petritis. Thermodynamics of a Brownian bridge polymer model in a random environment. *Journal of Physics A—Mathematical and General*, 29(6):1267–1279, March 1995.

[137] Matlab reference guide. The MathWorks, October 1992.

[138] D. McLaughlin. Recent developments in hydrologic data assimilation. *Reviews of Geophysics, Supplement: U.S. National Report to International Union of Geodesy and Geophysics 1991-1994*, pages 977–984, July 1995.

[139] D. McLaughlin. A reassessment of the groundwater inverse problem. *Water Resources Research*, 32(5):1131–1161, May 1996.

[140] D. Menemenlis, P. Fieguth, C. Wunsch, and A. Willsky. Adaptation of a fast optimal interpolation algorithm to the mapping of oceanographic data. *Journal of Geophysical Research*, 102(C5):10,573–10,584, May 1997.

[141] Y. Meyer. *Ondelettes et Opérateurs, I: Ondeletts, II: Opérateurs de Calderón-Zygmund, III: (with R. Coifman), Opérateurs multilinéaires.* Hermann, Paris, 1993. English translation of first volume, *Wavelets and Operators*, is published by Cambridge University Press.

[142] J. Ni, K. Ho, and K. Tse. Model-based multirate Kalman filtering approach for optimal two-dimensional signal reconstruction from noisy subband systems. *Optical Engineering*, 37(8):2376–2386, August 1998.

[143] L. Novak, G. Halversen, G. Owirka, and M. Hiett. Effects of polarization and resolution on the performance of a SAR automatic target recognition system. *Massachusetts Institute of Technology Lincoln Laboratory Journal*, 8(1), Spring–Summer 1995.

[144] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing.* Prentice Hall, Upper Saddle River, NJ, 1989.

[145] A. Oppenheim and A. Willsky. *Signals and Systems.* Prentice Hall, Upper Saddle River, NJ, 1997.

[146] O. Ore. *Graphs and Their Uses.* Random House, New York, 1991.

[147] A. Papoulis. Maximum entropy and spectral estimation: a review. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 29, pages 1176–1186, December 1981.

[148] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, second edition, 1984.

[149] B. Parlett. *The Symmetric Eigenvalue Problem*, chapter 15: The General Linear Eigenvalue Problem, pages 302–328. Prentice-Hall, Inc., 1980.

[150] D. Petris. Thermodynamics for the zero-level set of the Brownian bridge. *Communications in Mathematical Physics,*, 125(4):579–595, 1989.

[151] G. Picci. Geometric methods in stochastic realization and system identification. *CWI Quarterly*, 9(3):205–240, 1996.

[152] M. Porsani and T. Ulrych. Levinson-type extensions for non-Toeplitz systems. *IEEE Transactions on Signal Processing*, 39(2):366–375, February 1991.

[153] I. Primus. Scale-recursive estimation of precipitation using remote sensing data. Master's thesis, Massachusetts Institute of Technology, June 1996.

[154] R. Rao. The use and interpretation of principal component analysis in applied research. *Sankhya, series A*, 26:329–358, 1964.

[155] H. Rauch, F. Tung, and C. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965.

[156] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, pages 14–35, October 1991.

[157] D. Rose. Triangulated graphs and the elimination process. *Journal of Mathematical Analysis and Applications*, 32:597–609, 1970.

[158] D. Rose, R. Tarjan, and G. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computation*, 5(2):266–283, June 1976.

[159] N. Rozario and A. Papoulis. Spectral estimation from nonconsecutive data. *IEEE Transactions on Information Theory*, 33(6):889–894, November 1987.

[160] M. Schneider. Multiscale methods for the segmentation of images. Master's thesis, Massachusetts Institute of Technology, June 1996.

[161] M. Schneider. A Krylov subspace estimation algorithm. PhD thesis proposal, November 1998.

[162] M. Schneider, P. Fieguth, W. Karl, and A. Willsky. Multiscale methods for the segmentation of images. *IEEE Transactions on Image Processing.* To appear.

[163] M. Schneider and A. Willsky. Krylov subspace estimation. *SIAM Journal on Scientific Computing.* Submitted.

[164] M. Schneider and A. Willsky. A Krylov subspace method for large estimation problems. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, AZ.

[165] J. Schroeder and D. Howard. Multiscale modeling for target detection in complex synthetic aperture radar. In *Proceedings of the Asilomar Conference on Signals, Systems, and Signal Processing*, November 1998.

[166] F. Sellan. Synthèse de mouvements Browniens fractionnaires à l'aide de la transformation par ondelettes. *C.R.A.S. Paris Sér. I Math*, 321:351–358, 1995.

[167] J. Shah. Segmentation by nonlinear diffusion, II. In *Proceedings of IEEE Computer Society Conference on Vision and Pattern Recognition*, pages 644–647, 1992.

[168] K. Shanmugan and A. Breipohl. *Random Signals: Detection, Estimation, and Data Analysis.* John Wiley and Sons, Inc., 1988.

[169] M. Sironvalle. The random coin method: solution of the problem of simulation of a random function in the plane. *Mathematical Geology*, 12(1):29–36, January 1980.

[170] T. Speed and H. Kiiveri. Gaussian Markov distributions of finite graphs. *The Annals of Statistics*, 14(1):138–150, 1986.

[171] P. Stoica and R. Moses. *Introduction to Spectral Analysis.* Prentice-Hall, Upper Saddle River, NJ, 1997.

[172] G. Strang. *Linear Algebra and its Applications.* Saunders College Publishing, New York, 1988.

[173] G. Strang. Wavelets and dilation equations: a brief introduction. *SIAM Review*, 31(4):614–627, 1989.

[174] G. Strang. Wavelet transforms versus Fourier transforms. *Bulletin of the American Mathematical Society*, 28(2):288–305, April 1993.

[175] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, 1996.

[176] M. Tahir. A valuation formula for foreign currency options. In *Proceedings of International Federation of Automatic Control Symposium on Modeling and Control of National and Regional Economics*, Gold Coast, Qld., Australia, July 1995.

Appears in Modeling and Control of National and Regional Economies 1995. A Postprint Volume from the IFAC/IFIP/IFORS/SEDC Symposium. Pergamon. 1996, pp.463-6. Oxford, UK.

[177] R. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13(3):566–579, 1984.

[178] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–8(2):129–139, March 1986.

[179] A. Tewfik and M. Kim. Correlation structure of the discrete wavelet coefficients of fractional Brownian motion. *IEEE Transactions on Information Theory*, 38(2):904–909, March 1992.

[180] D. Tucker, May 1999. Private communication.

[181] A. Van der Veen, E. Deprettere, and A. Swindlehurst. Subspace based signal analysis using singular value decomposition. *Proceedings of the IEEE*, 81(9):1277–1308, 1993.

[182] C. Van Loan. *Computational Frameworks for the Fast Fourier Transform.* SIAM, 1992.

[183] P. Van Overschee and B. De Moor. *Subspace identification for linear systems.* Kluwer Academic Publishers, Norwell, MA, 1996.

[184] H. Van Trees. *Detection, Estimation, and Modulation Theory, Part I.* John Wiley and Sons, Inc., 1968.

[185] B. J. West and A. L. Goldberger. Physiology in fractal dimensions. *American Scientist*, 75:354–365, 1987.

[186] D. West. *Introduction to Graph Theory.* Prentice Hall, Upper Saddle River, NJ, 1996.

[187] J. White. Numerical algorithms. 6.336 Course Notes, Massachusetts Institute of Technology.

[188] A. S. Willsky and G. Wornell. Stochastic processes, detection, and estimation. 6.432 Course Notes. Massachusetts Institute of Technology.

[189] J. Woods. Two-dimensional discrete Markovian fields. *IEEE Transactions on Information Theory*, 18(2):232–240, March 1972.

[190] G. Wornell. Wavelet-based representations for the $1/f$ family of fractal processes. *Proceedings of the IEEE*, 81(10):1428–1450, October 1993.

[191] G. Wornell and A. Oppenheim. Estimation of fractal signals from noisy measurements using wavelets. *IEEE Transactions on Signal Processing*, 40(3):611–623, March 1992.

[192] G. Wornell and A. Oppenheim. Wavelet-based representations for a class of self-similar signals with application to fractal modulation. *IEEE Transactions on Information Theory*, 38(2):785–800, March 1992.

[193] B. Wu and Y. Su. The ergodicity analysis of two-dimensional discrete wavelet-based fBm fields. *IEEE Transactions on Signal Processing*, 46(3):805–809, March 1998.

[194] C. Wunsch. *The Ocean Circulation Inverse Problem.* Cambridge University Press, New York, 1996.

[195] B. Yazici and R. Kashyap. A class of second-order stationary self-similar process for $1/f$ phenomena. *IEEE Transactions on Signal Processing*, 45(2):396–410, February 1997.

[196] J. Zhang. The mean field theory in EM procedures for Markov random fields. *IEEE Transactions on Signal Processing*, 40(10):2570–2583, October 1992.

[197] J. Zhang. The mean field theory in EM procedures for blind Markov random field image restoration. *IEEE Transactions on Image Processing*, 2(1):27–40, January 1993.

[198] J. Zhang. The application of the Gibbs-Bogoliubov-Feynman inequality in mean field calculations for Markov random fields. *IEEE Transactions on Image Processing*, 5(7):1208–1214, July 1996.

[199] J. Zhang and G. Walter. A wavelet-based KL-like expansion for wide-sense stationary random processes. *IEEE Transactions on Signal Processing*, 42(7):1737–1745, July 1994.

Index